TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

CHENYU SHI

# NETWORK INTRINSIC QOE METRICS FOR VIDEO TRANSMISSION

Master's thesis

Examiner: Dmitri Moltchanov
Prof. Yevgeni Koucheryavy
Examiner and topic approved by the Faculty Council of the Faculty of Computing and Electrical Engineering on 6 March 2013.

# ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Degree Programme in Information Technology
**Chenyu Shi**: Network Intrinsic QoE Metrics for Video Transmission
Master of Science Thesis, 43 pages, 4 Appendix pages
June 2013
Major subject: Communication engineering
Examiner: Dmitri Moltchanov
Keywords: Video Streaming, video compression, video evaluation, peak signal-to-noise ratio, drop tail, active queue management, random early detection, packet loss, correlation, network intrinsic metrics

Nowadays , networking is more and more important to people's lives. Especially video streaming is playing a significant role in study and entertainment life. Many new applications appear to give people better videos. Because of high definition video, video compression and evaluation techniques become very useful to not only video web sites but also network operation and providers. Talking about video evaluation, Quality of experience (QoE) is an important indicator indicating the user experience of a video.

There are a number of factors affecting performance of video delivery in the Internet with queue management discipline being one of the most important. From the networking perspective, there are two main router queue management, drop tail and Active Queue Management (AQM). Drop tail is widely used and it is simple to configure and maintain. Even though it may cause continuous packet loss when congestion happens over the network which may have a great impact on video streaming, it, nowadays, is still widely used. AQM could be a better way to manage the router buffer. Random early detection (RED) is one of AQM and it can avoid congestion because it drops packets randomly. However, it is more difficult to configure.

The experiment in this paper is a statistical experiment to get the relationship between packet loss probability and correlation and QoE to provide a new network-intrinsic QoE metric..The process of video transmission over the network is simulated. The new metric is obtained by analyzing the obtained results. Even though it is a reference model, it is still very important. First, it gives a better way to estimate video quality using the network parameters. Second, analyzing the obtained results we see that , in order to get a better quality of video, RED is a better choice.

In the future, the more accurate metrics can be obtained by more times of experiment. Such values would provide more detailed quantitative relationship between packet loss probability and correlation and QoE.

## PREFACE

My thesis lasts about six months and during this time, many difficulties were met. Because of the help from my supervisor and my friends, all the problems were solved. I must say that my supervisor, Dmitri Moltchanov, is a really good supervisor. When I met some technical problems, he gave me much support. when I analyzed the experiment results, I was so confused at that time, since the data was so different from what I thought before the experiment was done and the data was in a mass.Dmitri Moltchanov gave me many suggestions on how to analyze data and graphs. In addition , he also gave me some hints on how to study in the field of network and programming, which, of course, helps me a lot in not only my thesis but also my study in the future. Moreover, I also want to say thanks to my friends and family, since they gave me much support.

Shi Chenyu
2013-06-14
Tampere

**CONTENT**

# 1. INTRODUCTION

Nowadays, people are becoming more and more dependent on the Internet for working, studying and entertainment, since the technology of the Internet is turning impossible into possible and making life and work more convenient and making entertainment more variable and interesting. It does not matter what we are talking about, e.g. working, studying or even entertainment in the Internet, multimedia first comes to our mind and video is the most significant part of multimedia. That is to say people spent much time in watching or skinning video in the Internet directly or download or have a video meeting and chat by online chatting applications like Skype and MSN. In addition, now people are not satisfied with short and low quality video, which leads bad user experience, and longer higher-definition video are demanding, which may cause much higher pressure on the network. So studying on video streaming or video transmission now is becoming more and more important.

## 1.1. Video Streaming

Because of huge amount of information in a video and limited throughput of a network, serious problems may be caused if the video is operated by computer as other kinds of file format. In general, only unbroken file can be played back by computer. That is to say, when some files need to be dealt with, it should be guaranteed that these files are unbroken. Once the files are broken or only half of the files is received, the files are thought to be broken by the computer and cannot be played back. If video file in the network is delivered by the method described above, the audience wait at least several minutes or even several hours for downloading the whole file, which, obviously, is unacceptable. So another method to handle this is needed. The new method is streaming technique, where we have a notion ‘bitstream’ instead of ‘file.’

### 1.1.1. Introduction to Video Streaming

The solution is a special streaming technique which is used to extract file. The basic idea of this streaming technique is that when a server is ready to transmit a video to a client, the video is divided into small chips (we still call it “frame” in order to be understood easily), which are similar to frame in image, according to chronological order by the server and then these frames are transmitted to the client. At the client side, these chips are played continuously by a certain player, even though there would be more or less delay which depend on the situation of the Internet. The idea above is also used by live Internet broadcast. In this situation, the new frames of video are generated

and transmitted immediately and continuously. In order to provide a reasonable user experience of displaying, a certain transmitting rate between server and client is needs to be guaranteed. If the transmition rate is very slow, the chips are displayed discontinuously. So, in practice, different Image definitions, according to transmitting rate and bandwidth in the network, should be provided. We see that there is no notion like a ‘file’ in this technique. This idea above is called streaming.

Video streaming is a method of transmitting video information. Client can begin to handle or display the video by this method, when only a part of the video is received at client side. That is to say, the client can receive and play the video simultaneously. It is a good way to solve the problem which is talked about above. The audience do not need to wait for a long time to watch a video, in addition, the audience may not even know that the video is still being transmitted. To sum up video streaming is a technology for distributing and watching digital video over the network.

The basic procedure of video streaming is video compressing/decompressing and video transmission. Video compressing is sometimes called encoding, while video decompressing is called decoding.

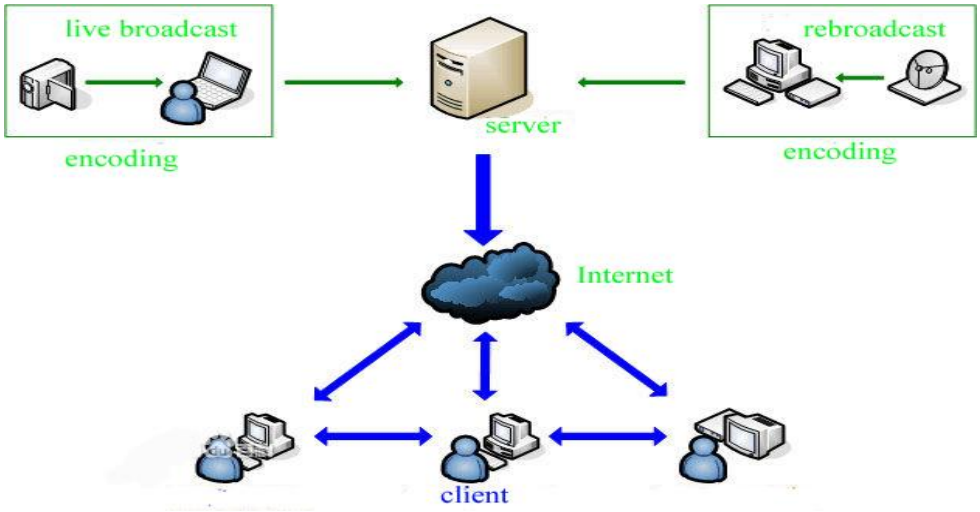



***Fig 1.1.*** *The process of video streaming*

As we know, video is often very huge, and if the video is transmitted directly without compression, huge pressure would be brought to the network and, if the pressure over the network is too huge, the network would be congested. So we need find a way to compress the file to save the bandwidth. Encoding is the method to handle this which is widely used nowadays. More details about encoding can be found in chapter 2. Moreover, decoding is the opposite of encoding and it is the method to decompressing the compressed video for further displaying [27;26;28;29].

### 1.1.2. Video Transmission and Associated Problems

Video transmission refers the process that encoded video frames are transmitted from server to client or client to client over the network. The network links may includes not only wired ones but also wireless links. Nowadays wireless access is more and more popular because it gives users more freedom and convenience, and it is more flexible, has good expansibility relatively low costs. However wireless network has also some disadvantages e.g. lower transmitting rate and lower stability, more security problems and higher maintenance costs. As we know, the wired and wireless network is still very fragile, although it has already been improved a lot by new modern technologies. Since there are so many factors to impact the network, ranging from human factors to natural technological ones. Because of these factors, the state of the network may become unpredictable and unreliable. Sometimes the distances between clients or between client and server is thousands of kilometers, and the packets containing video frames travel though many routers and many local area networks. Thus, the transmition rates and bandwidth can change dramatically between these local area networks which is decided by the situation of network. It has a great impact on the transmitting rate of the video and the size of the video packet. When the network becomes very crowded, the capacity of router buffer would be full or almost full and, at that time, packet losses would happen. It has influence on the quality of video. Since if the packets containing the video are dropped, that means some frames are lost, the video would be displayed discontinuously or distort greatly. That is to say some content in the video is lost, which brings very bad user experience to the audience.

The networking protocol supporting video streaming is another important issue which also has great influence on transmitting video. First, datagram protocols like the User Datagram Protocol (UDP) ,which is popular for transmitting video, send the video in time series of small packets. Although it is simple and efficient in the network, it brings a very serious problem. There is no mechanism in the protocol to guarantee successful delivery. The receiver has to detect packet loss or corruption and recover data by error correction techniques. If many of the video packet are lost, the video streaming may suffer a dropout, which may lead to losing or distorting content. While, reliable protocols, like transmission control protocol (TCP), guarantee correct delivery of each packet of video streaming, it accomplishes this by timeout and retransmission in the end-to-end manner. When data loss happens over the network, the sender may detect and retransmit the lost part and, of course, some delay may be introduced. It is still acceptable in video-on-demand scenarios, but in some case like video conference, server delay may bring bad user experience. Nowadays, HTTP/TCP are often used to transmit the control information, and RTP/UDP are used to transmit the video information. Second, the Real-time Streaming Protocol (RTSP), Real-time Transport Protocol (RTP) and the Real -time Transport Control Protocol (RTCP) were designed

specifically designed to stream media like video streaming over the network. RTSP can run over a variety of transport protocols, while RTP and RTCP are build on top of UDP. Another approach that seems to incorporate both the advantages of using a standard web protocol and the ability to be used for streaming live content, is adaptive bitrate streaming over HTTP. HTTP adaptive bit rate streaming is based on HTTP progressive download, but contrary to the previous approach, here the files are very small, so that they can be compared to the streaming of packets, much like the case of using RTSP and RTP. Third, unicast protocols is used to send copies of media stream from sender to different clients, but it does not scale well when the amount of users who want to watch the same video increases dramatically. In addition, even though, multicast protocols is designed to reduce the network loads of sending to different users, there is still a potential disadvantage of multicast protocols. It is the loss of video-on-demand functionality. Continuous streaming of video usually precludes the audience's ability to control playback. Last, Peer-to-Peer (P2P) protocols arrange for prerecorded streams to be sent between computers, which prevents the server and the network connections from becoming a bottleneck . However, it raises technical, performance, quality, and business issues.[26.]

Another issue is the buffer at the receiver. As we know, in the network, asynchronous transmission is used. The video is divided into several packets, and because of the instability of transmission, the routes of packets can be quite different. At the receiver, sequence of the received packets may be as the same as the time sequence of the video and sometimes some packets are even lost. So, at the receiver, a buffer is needed to store the arrival packets and modify them into the right time sequence, and then these packets are displayed by the player.

### 1.1.3. Video Streaming Applications

As we a told about earlier, video streaming or streaming media is playing a very significant role in people's life from daily life to work, from study to entertainment. Video conference and some video websites like YouTube are getting more and more popular. These video websites also try everything to make video display smother and smother and provide higher definition video. Nowadays, many applications for mobile terminals like Ipad are developed to satisfy the need that people want to watch videos when computer is not available for them. Moreover, as the service of Internet is becoming better and better, more and more audiences especially young audience like to watch TV programme on-line. There are several kinds of TV online like Internet Protocol television (IPTV) and Internet television (Internet TV or online TV). IPTV is a system through which television services are delivered using the Internet protocol suite over a packet-switched network such as the Internet, instead of being delivered through traditional terrestrial, satellite signal, and cable television formats.
One official definition approved by International Telecommunication Union is :

“IPTV is defined as multimedia services such as television/video/audio /text /graphics /data delivered over IP based networks managed to provide the required level of quality of service and experience,security, interactivity and reliability.”

Another definition of IPTV is given by Alliance for Telecommunication Industry Solution (ATIS) IPTV Exploratory Group on 2005:

“IPTV is defined as the secure and reliable delivery to subscribers of entertainment video and related services. These services may include, for example , Live TV, Video On demand (VOD) and Interactive TV (iTV).These services are delivered access agnostic, packet switched network that employs the IP protocol to transport the audio, video and control signals.In contrast to video over the public Internet, with IPTV deployments,network security and performance are tightly managed to ensure a superior entertainment experience, resulting in a compelling business environment for content providers, advertisers and customers alike.”

IPTV services may include the following: live television broadcast, with or without interactivity related to the current show; time-shifted television, which means to replay a TV show that was broadcast or replay a current TV show from its beginning; video-on-demand (VOD) which means to browse a catalog of videos, not related to TV programming. In addition people can play some games. IPTV get access to network by customer-premises equipment like set-top boxes which is an information appliance device that generally contains a tuner and connects to a television and an external source of signal, turning the source signal into content in a form that can be displayed on the television screen or other display device, and is also used to enhance source signal quality. The playback of IPTV requires an IP connected device like computer or smart phone, or the set-top box which is connected to a TV. Video content is often compressed by mostly, MPEG-4 (H.264) codec and then sent in an MPEG transport stream delivered via IP multicast in case of Live TV or via IP unicast in case of VOD. In addition, to VOD, UDP or RTP protocols for channel for streams and RTSP is used to transmit control information.

There are many advantages on IPTV. First, it make TV viewing experience more interactive and personalized. Audiences can share their commits with others who are also watching the same TV show and even can get feedback from the TV show. Second, people can synchronize their smartphones, computers and other devices. Third , IPTV brings VOD to user , which allows that audience can browse an online program or film catalog and select what they want to watch. Nowadays people often use VOD to watch high definition movies. Last, IPTV make our TV more than a TV, nowadays operating systems like Andriod or Linux is installed into modern televisions in order to get access to the Internet. User can install some applications on their TV, which brings even more intelligence. IPTV still has some disadvantages. The most serious problem is that it is so

sensitive to packet losses which may lead to the loss and distortion of the content and delay which may cause a long-time waiting. Nowadays, many companies try to develop a new IPTV application or add IPTV to their existing successful applications. For example, Xiaomi box is the latest IPTV device, and it is a set-top box but it is more intelligent and have more advanced functions. In addition, some traditional P2P video streaming products also add IPTV to their own product , like PPStream, and PPLive.
Internet TV is also an application of video streaming. It is the digital distribution of television content via the Internet. It allows the users to choose the content or television show they want to watch from an archive of content or from a channel directory. The two forms of viewing Internet television are streaming the content directly to a media player or simply downloading the media to the user's computer. And P2P video streaming, like PPStream, QQStream and some video communication application like Skype are all related to video streaming [24;25;27].

## 1.2. Network and Queuing Disciplines

As we know, network should provide the route for video transmission from source to destination. Some information of network like classic traffic, network-intrinsic metric and queuing disciplines of router is need to be introduced.

### 1.2.1. Classic Network and Traffic

The classic network contains elastic traffic and non-elastic traffic, which represents different methods for transmission. Elastic traffic is "network friendly", which means that the throughput or rate of network between end hosts or source and client is adjusted in response of changing of network situation. Elastic traffic is often TCP-based and it is for loss-free in-order delivery. The basic idea is that network congestion can leads to packet loss at routers. Packet loss causes TCP to start its congestion avoidance algorithm and reduce transmit rate over the network, since TCP interprets it as an evidence that the network experiencing congestion. It guarantees that the receiver can receive every packets even though it may suffer much delay, especially, when the congestion of network is server. Even though it provides a reliable way to transmit packets, there are still some disadvantages. For example, since retransmission depend mostly on feedback from the receiver, which is called acknowledgement packet (ACK) the ACK packets brings huge load to the network. And it is stateful , which means both the sender and the receiver have responsible for keeping track of the state of the communication session, and connection-oriented, which means an connection between the sender and the receiver is established before sending packets. So the rate would be a little bit slow and not that flexible.

While non-elastic traffic is opposite to elastic traffic and the throughput or rate of network between end hosts or source and client is not adjusted in response of changing of network situation. It is often unreliable since when the congestion happens in the

network, packets are just discarded by routers. That is to say, that there is no any mechanism to compensate for the packet loss. It is often used to transmit media streaming like video , audio streaming or VoIP. In addition, it is stateless which means that neither the sender or the receiver has an obligation to maintain the state of the communication session, and connectionless, which means that it is not required to establish a connection before sending packets. So it is more flexible and fast.

### 1.2.2. Conventional Metrics

When talking about quality of network, packet loss probability, which describes the probability of dropping packets, comes first. There are many reasons causing packet loss, for example signal fading in wireless networks, hardware failure, software corruption, and some interference from human and nature. But the main reason is that when buffer of router is full which is caused by the congestion of a network, the router would drop packets. In addition, packet loss has great effect on the transmitted data. To pure data, it produces data errors; in audio or video streaming, it causes jitter and frequent gap, during the video displaying or video conference, which gives very bad user experience.

Conventionally, when studying the quality or user experience of video streaming over Internet, the network-intrinsic metric is only the packet loss probability. It thought to be that the more lost packet are, the worse the video quality is. However, the conventional metric is not that accurate, since there are some other indictors which also have effect on the perceived quality.. As we know, the video is composed of thousands of static frames, and every static frames contains different information. Some of consecutive frames are very similar, while some consecutive frames may change very quickly. As we told above, it is unavoidable that some frames are dropped when the video is transmitting over the Internet. In one case, the dropped frames are totally separated, for example there are totally 300000 frames in the video, only one packet is dropped in every one thousand packets, and there are totally 300 dropped frames. In this case, at the receiver, the quality is acceptable, since the lost or distorted content is not continuous and it can be made up by the content from adjacent frames especially in the case, the adjacent frames are very similar.

To support our discussion, a small experiment is carried out. In this experiment, a simulation of the procedure of transmitting video streaming is carried out and then we compare the video at receiver and the original one at the sender using the peak signal-to-noise ratio (PSNR), which is a popular metric for estimation of the quality of video. The bigger PSNR is (usually 35-40), the better the video quality is. The details of PSNR metric is introduced in the next chapter. The video is short and is just about 300 frames. In the first setting, we dropped the packets separately, meaning that it was not possible to drop packets continuously and the totally dropped packets were 15. The

cortresponding PSNR was 34.6589 ,which means that the dropped packets had little to no effect on the quality of video.

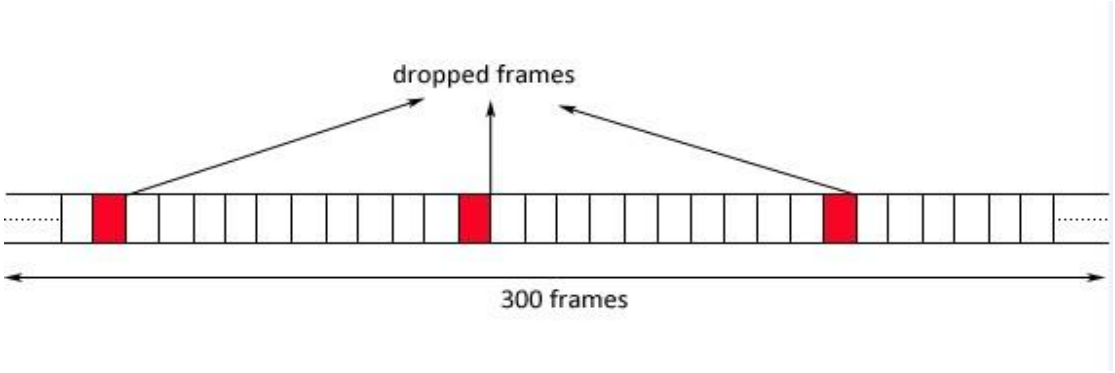



***Fig 1.2.*** *Dropped frames in low correlation*

In another setting, we also used the same video. The totally dropped frames were also 15, but they are continuous. In this case the PSNR was as low as 17.1884, which showed that the quality of the video is very bad.

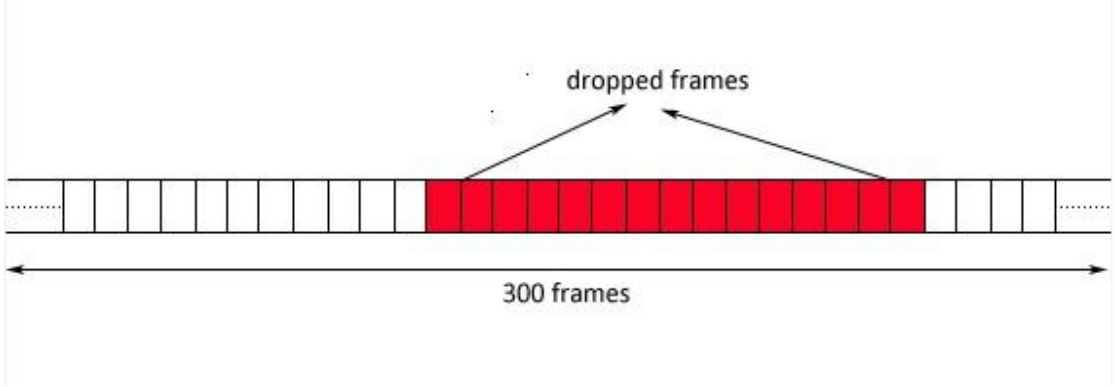



***Fig 1.3.*** *Dropped frames in high correlation*

From this small experiment, we can see that even the packet loss probability is the same, the PSNR is very different. As a result, the positions of the lost packets has great impact on the quality of video. So, packet loss is not the only indictor affecting the quality and user experience of video streaming, even though it is a very important one. Obviously, whether the dropped packets are continuous or not can have a great influence on the quality and user experience of video streaming.

The described dependence is in fact correlation. In statistics, correlation expresses the amount of linear dependence between random variables. In our situation, the more continuous the dropped packet are, the higher the correlation is. In other words, it represents the "contiunity" of packet losses. Now, we can safely claim that correlation of the packet loss process may also has influence to the quality and user experience of video streaming.

### 1.2.3. Queuing Disciplines

In this part, the queuing disciplines used inInternet routers are introduced. There are several queuing disciplines. The traditional way is called drop tail or tail drop and belongs to a class of passive queue management (PQM) disciplies, while the latest way is called active queue management (AQM). Drop tail resembles characteristics of widely known first-in first-out (FIFO) queuing. The basic idea of FIFO is that the sooner thee packet arrives to a router the sooner it should be transmitted to the next hop. That is, the router handles and forwards packets according to their arrival sequence. It is very easy to implement, which is the most obvious advantage of FIFO. However, the most disadvantage is that a router does not take into account the importance of a packet when it forwards it, eventually leading to delays and jitter. It is terrible for some services which are sensitive to delay and jitter like video streaming and video conferencing. In addition, an improved FIFO is queuing with priority. The idea is that different priorities are assigned to different kinds of packets. At the router, there are several queues and each queue is map to a certain priority and in each queue, we still use FIFO. It decreases delays and jitter of those packets whose priority is high [21].

Drop tail is very similar to FIFO. It is a queue management algorithm used by routers to determine when to drop packets and when to begin to receive packets again. When using drop tail all the arriving packets is treated similarly. Generally, packets stand in a queue at the buffer of a router when they arrive at the router. When the buffer of a router is filled to its maximum capacity, the new arrival packets are dropped continuously until the buffer has enough room to receive the incoming packets. The most obvious advantage of drop tail is that it is very easy to implement, while the disadvantage is that it may cause great delays, congestion and jitter for delay sensitive traffic like media streaming. In addition, it also causes an avalanche of dropped packets when the buffer is full, which gives very bad quality and user experience video to the audience. In the improved priority-based implementation, it brings different priorities to different packets and there are several queues for different priorities at the router buffer. The router handles the packets of high priority first. It also solves the problem of packet losses and delay to some extent. Since it is easy to implement, drop tail is still widely used over nowadays network. However, it is not so good to modern networking services, which includes media streaming. First the continuous packet dropping may lead to high correlation of the packet loss process, which is discussed about above. Higher correlation of video streaming can cause loss of some chips when displaying the video. Second it may cause great delay when the buffer capacity is too big. Third, if the buffer is very short, it may cause network congestion, especially when the bandwidth and transmitting rate is low [21,23]**.**

Active queue management (AQM) is a new technique that is built on top of intelligent packet dropping or explicit congestion notification (ECN) marking packets before a

router's queue is full. It operates by maintaining one or more dropping probabilities, and probabilistically drops packets even when the queue is not full. The advantage of AQM is that it provides a shorter queue and, as a result, reduce network congestion and delay. Random early detection (RED) also called random early drop or random early discard, is an active queue management algorithm. The goal of RED is to avoid congestion , global synchronization and bursty traffic and to limit delay by bounding the average queue length. The basic idea is that RED monitors the average queue size and drops packets probabilistically based on certain thresholds. If the buffer is almost empty, all the incoming packets are accepted. As the queue grows, the probability for dropping an incoming packet grows too. During this time, some packets begin to be dropped. At last, when the buffer is full, that is the probability of dropping reaches one and all the arrival packets are dropped. The basic implementation steps are:

1. Define the two thresholds for the buffer queue size $TH_{MIN}$ and $TH_{MAX}$
2. When a packet arrives, compute the average queue size $AVG$ ;
3. If $AVG < TH_{MIN}$ queue packet (no congestion);
4. Else if $AVG > TH_{MAX}$ drop this packet (severe congestion);
5. Else drop this packet with probability $P_a$ (raising congestion) [22;23]**.**

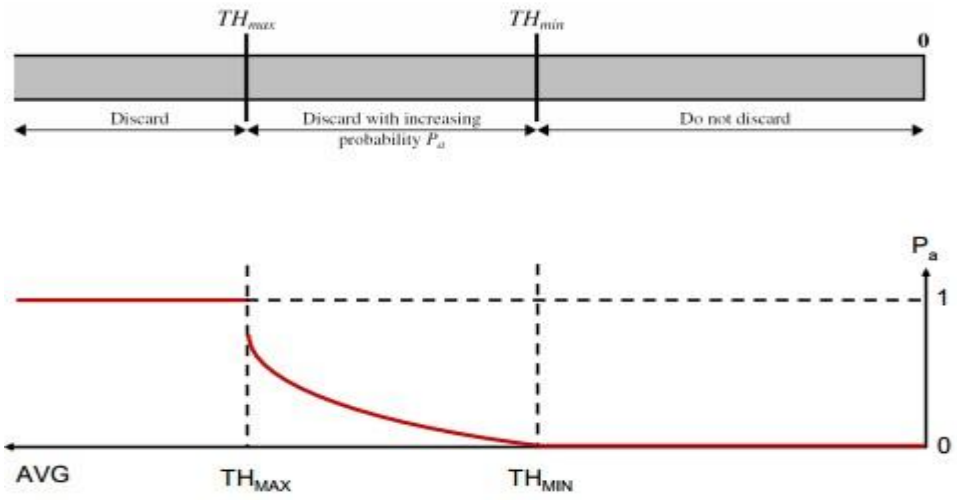



***Fig 1.4.*** *RED in detail***[23.]**

### 1.2.4. Summary

We often use UDP to transmit media streaming, since it is fast and there is no need to construct a route before transmitting data. However, it is not reliable and there is no any mechanism to remedy the dropped packets, which is terrible to the transmission, especially, for the video streaming.

Conventionally packet loss probability is the only factor for estimating the quality of video. The higher the packet loss probability is, the worse the quality of video is. At the same time, sometimes, even when the packet loss probabilities are the samethe video quality differs. So there must be some other factors that have influence on it. We demonstrated that correlation is one of such indicators.

As we discussed in Section 1.2.3, drop tail is an easy way to manage the buffer of a router, and it is used widely in the Internet. However, there are many drawbacks of this scheme including severe packet losses, long delays and jitter. In media streaming, drop tail may cause high correlation of the packet losses leading to quality deterioration of the displayed video. However, there are some other new methods to manage the queue at router. RED is one of them. It begins to drop packets when the queue size is not full preventing the network from deep congestion. The probability of packet dropping grows as the queue grows. It could be really good for video streaming , since less content is lost when displaying the video, even though the packet loss probability is very high [2].

## 1.3. Formulation of the Problem

According to section 1.2.2, the conventional network-intrinsic metric for streaming video is the packet loss probability. However, the packet loss probability in isolation provides incomplete information, as, sometimes , videos containing the same packet loss probability may lead to totally different PSNRs. In addition, the most widely used queue management discipline used in the Internet is drop tail. It may cause continuous packet losses for video streaming service which may result in severe quality distortion.

According to the abovementioned explanation, in order to give a more accurate evaluation of network and video streaming, correlation should be taken into account. In this thesis, a new network-intrinsic metric combining the packet loss probability and correlation is introduced. According to the experiments conducted in this thesis, AQM should be used in the network, e.g. RED. It begins to drop packet before the length of queue reaches the maximum capacity of the buffer. One advantage, especially to video streaming, is that it may would result in the packet loss process characterized by low correlation or no correlation at all. This would improve PSNR as was demonstrated in the experiment discussed in Section 1.2.2.

More details about the new network-intrinsic metric combining packet loss probability and correlation and the advantages of RED will be explained in next few chapters.

# 2. VIDEO COMPRESSION AND QUALITY EVALUATION

Video compression is an essential procedure of video transmission which may have a great impact on it. Some knowledge and information on video compression need to be introduced in this chapter. In addition, the common method of video quality evaluation like PSNR is also introduced here.

## 2.1. Video Compression

In this section, the detailed information of video compression is provided. First the basic information of video compression are introduced. Second, the types of video redundancy removed by a lossy compression procedure are discussed. Third, some codecs are also introduced.

### 2.1.1. Introduction to Video Compression

Video compression in the nutshell is encoding video information using fewer bits than the original one in order to transmit and save video easily The uncompressed (raw) digital video signal needs a very high bandwidth, which is usually well above 20MB/s. In this case, it is very difficult to transmit a video file and the amount of required bandwidth would be too large for a network to handle. The bandwidth can be decreased to 1-10MB/s by video compression technique. Thus, after encoding the video, it is much easier to transmit it.

Video compression usually contains an encoder and a decoder. Encoder is used to convert original video into an certain compression format, in order to transmit or store. Decoder converts compressed format into the original video (uncompressed one). The combination of an encoder and a decoder is called codec or encoder/decoder.

Compression can be either lossy or lossless. In lossless compression, no information is lost. Lossless data compression algorithms usually exploit statistical redundancy (discussed in next session) to represent data more concisely without losing any information. It is indeed possible since most real-world data has statistical redundancy. For example, an image may have areas of color that do not change over several pixels, and instead of coding “read pixel, red pixel, ...” the data may be encoded as “200 red pixels”. There are many schemes to reduce size by eliminating redundancy, e.g. Huffman compression. I have to say that the compression rations are not huge for these

schemes. In lossy compression schemes, some loss of information in the video is acceptable. For example, dropping nonessential detail from the video source can save storage space and transmission bandwidth. Lossy video compression schemes are inspired by psychovisual perception research telling us how people perceive the video in question. For example, the human eye is more sensitive to subtle variation in luminance than it is to variations in color. Lossy compression is a trade-off between the lost information and the size reduction. In practice, the majority of video compression algorithms are lossy. The results of lossy compression is much better than that of lossless compression, even though, in lossy compression, some information is lost, which, of course, has little effect on displaying. In all compression algorithms, there is a trade-off between video quality, cost of processing the compression and decompression, and system requirements.

Some highly compressed video may present visible or distracting artifacts. Video compression typically operates on square-shaped groups of neighboring pixels, often called macroblocks. These pixel groups or blocks of pixels are compared from one frame to the next, and the video compression codec sends only the differences within the blocks. In areas of video with more motion, the compression algorithm must encode more data to keep up with the larger number of pixels that are changing. Commonly during explosions, flames, flocks of animals, and in some panning shots, the high-frequency detail leads to quality decreases or to increases in the variable bit rate [1,20].

### 2.1.2. Redundant Video Information

There is a high correlation in the video data, that is, there is a very huge amount of redundant information in the video. Video compression technique is implemented by reducing the redundancy information. The redundancy information is classified as temporal redundancy, spatial redundancy, statistical redundancy, and perceptual redundancy. Temporal redundancy means that there is high correlation between the adjacent frames of video ,which is called temporal redundancy. Video compression between frames like motion compensation, motion representation and motion estimation can reduce the temporal redundancy. Motion compensation is a technique to predict and compensate a certain part of the current frame according to the same part on previous frame. It is very useful to reduce redundancy. Different motion vectors can describe motion information in different parts of frames. The motion vector can be compressed by entropy encoding. Motion estimation is a packet of techniques to extract motion information from video streaming. Spatial redundancy means that there is high correlation between neighbor pixels in a certain frame. The correlation is called spatial redundancy. In order to reduce the redundancy, intra-frame coding like transform coding, quantization coding, and entropy coding which are the lossless coding method, are mainly used. Transform coding is used to transform spatial signal to another orthogonal vector space, which reduces the correlation and redundancy, since there is huge amount of redundancy in intra-frame. After transform coding, a couple of parameters are generated and then these parameters are quantized. The process is called

quantization coding. Statistical redundancy means that the distribution of pre-coding symbol is nonuniform. Lossy video compression is achieved by reducing the statistical redundancy. Perceptual redundancy refers the information that is barely observed by the audience, i.e. details in the corners of the picture.

The following is the description of the basic process of reducing redundancy. When encoding the current signal, a signal which is a prediction of the current one (called predicted signal) is generated by the codec. The method of prediction can be inter prediction, which is predicted according to previous frame, or intra prediction, which is predicted by neighbor pixels in the same frame. Then, after generating the predicted signal, a signal called residual signal is generated by substracting predicted signal from current signal. The residual signal is the one that is encoded. In this case, a part of temporal redundancy and most spatial redundancy is eliminated. Codec then transforms residual signal using discrete cosine transform (DCT) and then quantize it in order to eliminate more temporal redundancy and spatial redundancy instead of encoding residual signal directly. The parameters of quantization are encoded by entropy coding, which can reduce statistical redundancy. At receiver, a similar and converse way is operated to get the reconstructed video [19].

### 2.1.3. Video Codecs

A video codec is a device or software that enables compression or decompression of digital video. Historically, video was stored as an analog signal on magnetic tapes. Around the time when the compact disc entered the market as a digital-format replacement for analog audio, it became feasible to also begin storing and using video in digital form, and a variety of such technologies began to emerge.

Nowadays, lossy compression codecs are widely used. For example, videos is compressed by different kind of codecs in DVD (MPEG-2), VCD (MPEG-1), TV broadcast and Internet etc. In order to scan or display videos correctly, user need to download a packet of codecs, which are decoding or encoding applications for computer.

Here are some codecs standards and some typical codecs below.

- MPEG-1 part two

It is a codec standard mainly used in VCD and some online videos. It is similar to Video Home System (VHS) ,but it has higher coding rate and higher quality in a long time. MPEG-1 codecs have better compatibility and generality. That is to say, the video in MPEG-1 format can be displayed in almost all the computers. MPEG-1codcs scan the video raw by raw only.

- MPEG-2 part two

It is mainly used in DVD, Blue-ray and some online video. It provides higher quality and widescreen presentation. In addition, MPEG-2 codecs can scan the video by every

two rows (interlaced regime). Even though, they are very old codecs, MPEG-2 codecs are widely popular and acceptable nowadays.

- MPEG-4 part two

It is mainly used in video transmission over network, broadcast and media saving. It is characterized by a higher compression ratio and uses an object-oriented encoding method. Both scanning row by row and scanning every two rows are supported by MPEG-4 codecs. FFmeg is one of the famous MPEG-4 part two codecs. It includes in the open-source libavcodec codec library, which is used by default for decoding or encoding in many open-source video players, frameworks, editors and encoding tools such as MPlayer, VLC, ffdshow or GStreamer. It is compatible with other standard MPEG-4 codecs like Xvid or DivX Pro Codec.

- H.261-H.263

H.261, H.262 and H.263 codecs standards are developed by ITU Telecommunication Standardization Sector (ITU-T, [18]) , which is one of the three sectors (divisions or units) of the International Telecommunication Union(ITU) and coordinates standards for telecommunications. These codecs are outdated now.

- H.264/MPEG-4 AVC

It is also called MPEG-4 part ten standard, which is the same as ITU-T H.264. It is the latest lossy video encoding technique developed by Video Coding Experts Group (VCEG or ITU-T VCEG) and Moving Picture Experts Group (MPEG). It is used more and more widely. Many applications are currently using these codecs, for example: PSP, Mac OS X v10.4, and Blue-ray. The bit rate is 10’s to 100’s kb/s [1]. Here are some other H.264 codecs:

  - X264: it is a GPL-licensed implementation of the H.264, but it is only an encoder.
  - QuickTime : it is an application of Apple.
  - Nero Digital: it is a commercial MPEG-4 and AVC codecs developed by Nero AG [17]**.**

- AVS

It is developed by Audio Video coding Standard Workgroup of China. It is not only the standard of video coding as it provides a different way of authorization to avoid huge amount of liscensing fees. In techniques, some parts of techniques of AVS are similar with that of H.264 [16]**.**

- WMV

Windows Media Video (WMV) is video codec family of Microsoft including WMV7, WMV8, WMV9, WMV10. They are used widely for all kinds of situations of network [17].

## 2.2. Video Quality Evaluation

As video streaming is becoming more and more popular and is taking important part of the daily life to most people. At the same time, new additional and various forms of video media contents are produced and are delivered through the network. The future of network is, surely, to be media oriented. Because of the development of video streaming, there is a profound need for an efficient user experience - quality of experience (QoE). QoE will become the prominent metric to consider when deploying networked media services.

Media service providers are more and more becoming interested in evaluating the performance of their services as perceived by the end users. In addition, network operators are also interested in evaluating QoE metics because it is very helpful to optimize the network and configure or reconfigure the network parameters to provide better user experience. However the video quality of end user, that is QoE, is very hard to estimate since it a subjective metrics, which is very hard to compute.

In this section, some video quality and QoE estimation metrics are introduced. We also discuss their advantages and disadvantages.

### 2.2.1. Quality of Experience

Quality of Experience (QoE) is defined as " the overall acceptability of an application or service, as perceived subjectively by the end user "[6]. It is different from network quality of service (QoS).The latter are known to be not sufficient to get a precise idea about the quality of video streaming. QoS cares about all aspects of network connection for example service response time, packet loss, signal-to-noise ratio, cross-talk, echo, interrupts, frequency response and so on. It is more "network-oriented", that is to say, as it cares more about networking situation instead of user experience. QoE focuses on the whole experience of the user from the source to the receiver, including the video itself and the network performance. In general, there are two approaches to measuring QoE. These are subjective assessments and objective assessments.

Subjective assessments consist of a panel of human being rating series of short video streaming according to their own personal view about quality [7]. QoE metrics predicted by a user are sensitive to the estimation error of arrival process [8]. They give the evaluation to the video from the end users, which sometimes are a little bit subjective and sensitive to test environment and the result may be very surprised. They are based on human judgment without any objective criteria except for overall impression of the service. However, subjective metrics are extremely useful for developing objective criteria of quality assessment. In general, there are two types of

subjective assessments. One of them is called quality assessments which establish the performance of the system under optimum conditions. Another one is called impairment assessments which establish the ability of system to retain quality under a not good conditions. In order to get a reasonable outcome of subjective assessments, an appropriate assessment must be selected according to the selected objective assessments and environments. The results of these tests is called a Mean Opinion Score (MOS) . MOS is a test that has been used to obtain the quality of network from the human user's view in telephony networks. It provides a numerical indication of perceived quality from the users' perspective of received media after compression and transmission and is expressed as a number in the range 1 to 5, where 1 represents the lowest video quality and while 5 represents the highest video quality. Since the preparation and execution of subjective tests is costly and time consuming, a lot of efforts have been focused on developing cheaper, faster and easier applicable objective assessments [3,15].

***Table 2.1.*** *Mean Opinion Score(MOS)*

| **MOS** | **Quality** | **Impairment** |
|---|---|---|
| 5 | Excellent | Imperceptible |
| 4 | Good | Perceptible but not annoying |
| 3 | Fair | Slightly annoying |
| 2 | Poor | Annoying |
| 1 | Bad | Very annoying |

Objective evaluation are algorithms and formulas that measure, in a certain way, the quality of a video streaming. Usually, these are point parameters. These parameters are often obtained by comparing the original and the transmitted or received video. The objective video quality metrics range from the very simple to the very complex ones. For example those metrics based on human vision systems tend to be very complicated. In addition, the original video is hard to obtain at end user's end, especially, when the receiver is very far from the source. Further, even when both original and the received video are both available, the measurement still cannot be done in real-time. The objective quality metrics can be further classified in three different categories: full-reference (FR), reduced-reference (RR) and no-reference (NR), based on the amount of information available for comparison with the original video:

- FR metric: original and distorted videos are available. Usual metrics are Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM);
- RR metric: distorted and some extracted structural features. Example of features: localized spatial and temporal activity;
- NR metric: only the distorted videos are available.

In our experiments in next chapter, FR metric is used to estimate the QoE of video streaming [3].

### 2.2.2. PSNR and SSIM Indices

In objective assessments of QoE, peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) are the common criteria. Both have their advantages and shortcomings. PSNR is the ratio between the maximum possible value (power) of a signal and the power of distorting noise that affects the quality of its representation [11]. PSNR ,which is a statistical criterion estimated per each individual frame and averaged over all the frames [5].. Although there is a huge deviation between PSNR and subjective assessments to some frames, PSNR is still very accurate to most frames and video sequences. PSNR is defined as:

$$PSNR = 10\log_{10}(\frac{a_{\max}^2}{MSE})$$

where MSE (mean square error) is defined as:

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j-1}^{N}[a(i,j)-b(i,j)]^2$$

In the formula, $a(i,j)$ and $b(i,j)$ are the corresponding grey values or color values of the original frame and reconstructed frame, respectively. $MN$ is the total pixel values of an $M \times N$ size frame. Next, $a_{\max}$ is defined as: $a_{\max} = 2^l - 1$, where $l$ is called color depth, which means the binary digit occupied by a pixel. In general, $l = 8$. The PSNR value approaches infinity as the MSE approaches zeros, and it means that a higher PSNR value provides a higher image quality. At the other end of the scale, a small value of the PSNR implies high numerical differences between images. The PSNR value of every frame in the video sequence is calculated individually. Then the PSNR value of video sequence is obtained as the average of PSNRs of all frames. The obvious drawback is that PSNR and MSE is a kind of computation of grey values and it ignores the effect, which is generated by the content of video to user's eyes. So it sometimes fails to give an accurate measurement to video quality [4,12]**.**

Another objective video assessments based on structural distortion. The reason behind is that HVS is good at extracting the structural information from the frames, and the measurement of the change of structural information is quite similar or close to the perception of the change of frame quality. So, the structural similarity is the evidence that the change between precious frame and current one is not obvious, that is, the loss of quality is low. Structural similarity index (SSIM) is an objective method that is very close to the perception of frame distortion. Mathematically, SSIM is based on the following idea. If the original frame is represented by a spatial vector, and any distorted frame can be represented by the original frame vector adding distortion vector. When the lengths of distortion vector and the original frame vector are equal, they can be defined on the sphere whose radius is MSE. . SSIM is designed by modeling any image distortion as a combination of three factors that are the loss of correlation, luminance distortion and contrast distortion. SSIM is defined as:

$$S(x,y) = f(l(x,y),c(x,y),s(x,y))$$

where $x$ and $y$ are the original and distortion frame respectively. $S(x,y)$ is the similarity between distortion signal and original signal and it is distortion measurement, $l(x,y)$ is luminance comparison function, $c(x,y)$ is contrast comparison function, $s(x,y)$ is structural comparison function. They are independent and $f(\ .\ )$ is a integration function. Here are the three comparison functions definition:

$$l(x,y)=\frac{2\mu_x\mu_y+C_1}{\mu_x^2+\mu_y^2+C_1}\ ,\ C_1=(K_1L)^2$$

$$c(x,y)=\frac{2\sigma_x\sigma_y+C_2}{\sigma_x^2+\sigma_y^2+C_2}\ ,\ C_2=(K_2L)^2$$

$$s(x,y)=\frac{2\sigma_{xy}+C_3}{\sigma_x\sigma_y+C_3}$$

$$\sigma_{xy}=\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\mu_x)(y_i-\mu_y)$$

where $L$ is the dynamic change of pixels, $K_1, K_2 << 1$, luminance average $\mu_x$, $\mu_y$ are luminance evaluation, and standard deviation $\sigma_x, \sigma_y$ are contrast evaluation. According to the formulas above, another expression of SSIM can be written as:

$$SSIM(x,y)=[l(x,y)]^{\alpha}\cdot[c(x,y)]^{\beta}\cdot[s(x,y)]^{\gamma}$$

In the formula, $\alpha, \beta, \gamma > 0$ and the three parameters is the modulation. Fig. 2.1 shows the framework of SSIM. In spite of a better conceptual foundation compared to PSNR, the results of SSIM are still different to the result of subjective assessments [4,5].

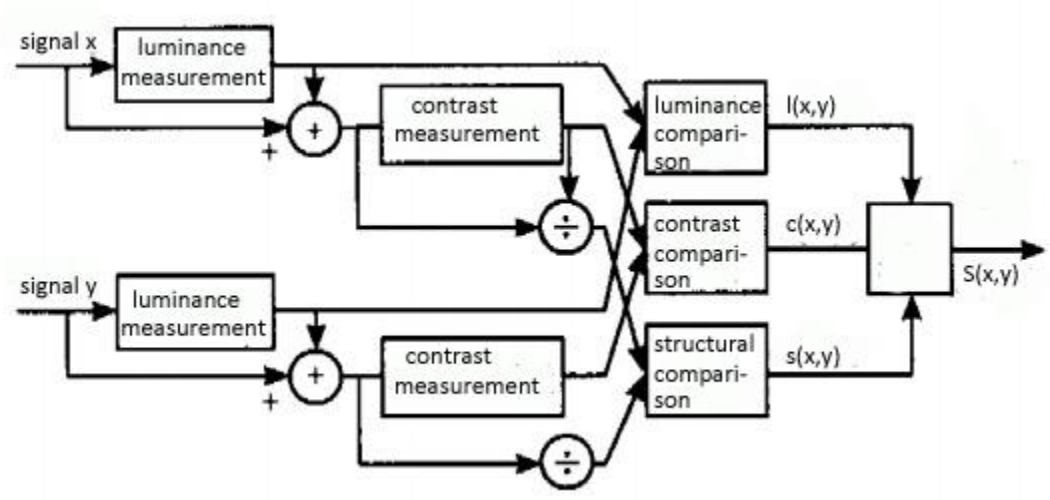



***Fig 2.1.*** *Framework of SSIM*

Nowadays, there are many modified methods based on PSNR and SSIM quality metric, which overcome the drawbacks of PSNR and SSIM. But all they still fail to bring reasonable results.

### 2.2.3. Perceived Quality Transmission

QoE is often evaluated at application layer. That is, QoE only cares about the original and distorted videos themselves. Even though QoE nowadays, is used as an indicator to optimize network services and configure or modify the parameters of a network, according to QoE, the evaluation of video transmission or video streaming over network is not that appropriate. The reason is that both PSNR and SSIM involves no parameters that are directly related to the network itself. Analyzing QoE results, we only know whether the networking is good or not. If the networking is very bad, the reason for such a bad performance is hidden Recall that video transmission is influenced by the complicated network situations where conventional performance metric such as packet loss, jitter, delay all degrades. As we discussed in Chapter 1 packet loss probability and loss correlation are the most significant factors affecting video quality. For video streaming over network, QoE and conventional network-intrinsic metrics are not enough at all. The way how packet loss probability and correlation impact QoE and the relationship between these factors are discussed in the following chapters.

# 3. SETUP OF EXPERIMENT

In this chapter, the set of experiments to explore the relationship between QoE and network-intrinsic metrics are described. These experiments are statistical in nature which simulates the video transmission over the network using MATLAB modeling environment. The basic steps are: video encoding, injecting random sequence to generate packet losses with a predefined loss probability and correlation, video decoding and PSNR calculation. Figure 6 is the working flow of this experiment. In addition to MATHLAB we also use C to generate random sequences. The steps and the idea will be discussed. The detailed results will be given in next chapter.

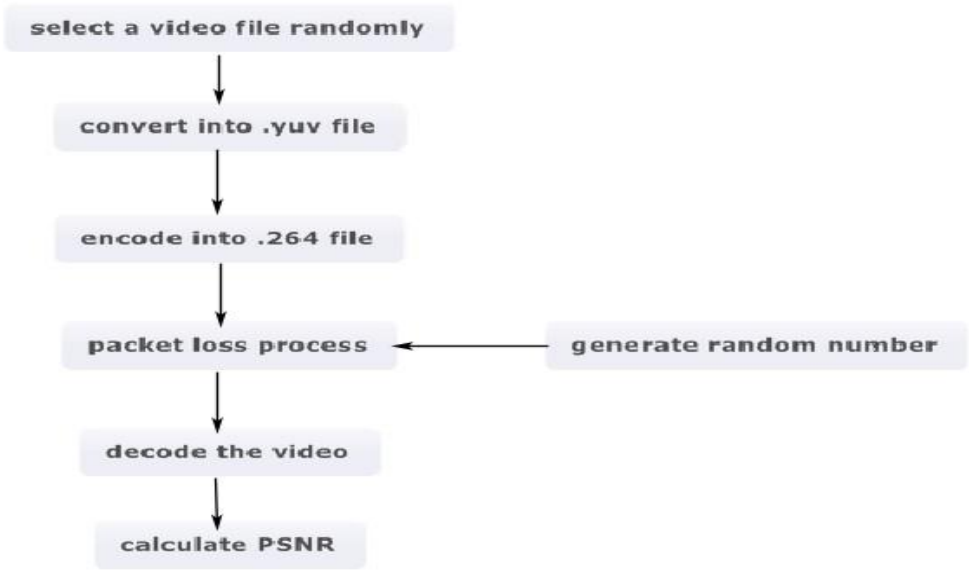



***Fig 3.1.*** *The flow of experiment*

In our experiments, three videos are selected, whose lengths are different , 32 minutes , 36 minutes and 42 minutes, respectively. Their titles are “short.avi”, “medium.avi”, “long.avi”. All of them are first converted into “.264” file.

## 3.1. Software Introduction

In this section, the software used in our experiments are introduced. In particular, we discuss JM (H.264/MPEG-4 AVC Reference Software) and ffmpeg. The latter is the set of H.264/MPEG-4 codecs.

### 3.1.1. ffmpeg Software

FFmpeg is a famous leading multimedia framework, which is able to encode, decode, transcode, mux, demux, stream, filter, and play almost anything that human and machines have created. It supports most nowadays formats, which were designed by video standards committee, the community or a corporation. It includes many libraries: libavcodec (the leading audio/video codec library), libavutil, libavformat, libavfilter, libavdevice, libswscale, and libswresample, which can be used by many applications. Ffmpeg contains ffserver, ffplay and ffprobe that can be used by the end users for transcoding, streaming, and playing. ffmpeg is free software licensed under the LGPL and GPL.

FFmpeg provides various tools.

These are

- ffmpeg is a command line tool to convert multimedia files between formats.
- ffserver is a multimedia streaming server for live broadcasts.
- ffplay is a simple media player based on SDL (Simple DirectMedia Layer) and the FFmpeg libraries.
- ffprobe is a simple multimedia stream analyzer.

In our experiments, ffmpeg is used to convert the downloaded video files, which are in the ".avi" format, into ".yuv" format files. The reason is that the format of input file for JM software is ".yuv".. ffmpeg is a convenient video and audio converter that is also used in convert a live video or audio source. It can convert between arbitrary sample rates and it can also resize video on-the-fly by a high quality polyphase filter. The basic synopsis is:

```
ffmpeg[global_opinion]{[input_file_opinions]-i'input_file'}...{[
output_file_opinion]'output_file' }...
```

ffmpeg reads one or more input files, specified by the `-i` option, and writes to one or more of output files, which are specified by a certain and plain output filename. Ffmpeg considers anything on the command line which cannot be interpreted into option as an output filename. In principle, the input and output file can contain any number of streams in different types which can be video, audio and so on. The limitation of the number or type is at the container format. In addition , selecting which streams or types from which inputs will go into which output can be done automatically or by some certain option. Moreover, if there are more than one input files, indices can be used to mark their options. Usually, options are applied to the next specified file. Same options can be used multiple times in one command line. Each occurrence is then applied to the next input or output file.

In our experiments we use the following command:

```
ffmpeg -i short.avi -s 320*240 short.yuv
```

The `short.avi` is the input file, and `short.yuv` is the output file. The `-s` option is aim to set the frame size of output file. Fig. 7 and Fig. 8 give the running environment of ffmepg. We use this software to get some ".yuv" files, which are unencoded videos [14].

```
e:\thesis\ffmpeg\bin>ffmpeg -i short.avi -s 320*240 short3.yuv
ffmpeg version N-51352-g81e85bc Copyright (c) 2000-2013 the FFmpeg developers
  built on Mar 27 2013 19:17:51 with gcc 4.8.0 (GCC)
  configuration: --enable-gpl --enable-version3 --disable-w32threads --enable-av
isynth --enable-bzlib --enable-fontconfig --enable-frei0r --enable-gnutls --enab
le-libass --enable-libbluray --enable-libcaca --enable-libfreetype --enable-libg
sm --enable-libilbc --enable-libmp3lame --enable-libopencore-amrnb --enable-libo
pencore-amrwb --enable-libopenjpeg --enable-libopus --enable-librtmp --enable-li
bschroedinger --enable-libsoxr --enable-libspeex --enable-libtheora --enable-lib
twolame --enable-libvo-aacenc --enable-libvo-amrwbenc --enable-libvorbis --enabl
e-libvpx --enable-libx264 --enable-libxavs --enable-libxvid --enable-zlib
  libavutil      52. 22.101 / 52. 22.101
  libavcodec     55.  2.100 / 55.  2.100
  libavformat    55.  0.100 / 55.  0.100
  libavdevice    55.  0.100 / 55.  0.100
  libavfilter     3. 48.105 /  3. 48.105
  libswscale      2.  2.100 /  2.  2.100
  libswresample   0. 17.102 /  0. 17.102
  libpostproc    52.  2.100 / 52.  2.100
Input #0, avi, from 'short.avi':
  Duration: 00:34:16.40, start: 0.000000, bitrate: 134 kb/s
    Stream #0:0: Video: tscc (tscc / 0x63637374), rgb555le, 1024x768, 15 tbr, 15
 tbn, 15 tbc
    Metadata:
      title           : Camtasia Producer_render7f141fa.avi 视频 #
    Stream #0:1: Audio: mp3 (U[0][0][0] / 0x0055), 24000 Hz, stereo, s16p, 56 kb
/s
    Metadata:
      title           : Microsoft Waveform: Camtasia Producer_convert7f41e37.wav

Output #0, rawvideo, to 'short3.yuv':
  Metadata:
    encoder         : Lavf55.0.100
    Stream #0:0: Video: rawvideo (RGB[15] / 0xF424752), rgb555le, 320x240, q=2-3
1, 200 kb/s, 90k tbn, 15 tbc
    Metadata:
      title           : Camtasia Producer_render7f141fa.avi 视频 #
Stream mapping:
  Stream #0:0 -> #0:0 (camtasia -> rawvideo)
Press [q] to stop, [?] for help
frame=   36 fps=0.0 q=0.0 size=    5400kB time=00:00:02.40 bitrate=18432.0kbits/
frame=   77 fps= 77 q=0.0 size=   11550kB time=00:00:05.13 bitrate=18432.0kbits/
frame=  120 fps= 79 q=0.0 size=   18000kB time=00:00:08.00 bitrate=18432.0kbits/
```

***Fig 3.2.*** *The running environment of ffmpeg(1)*

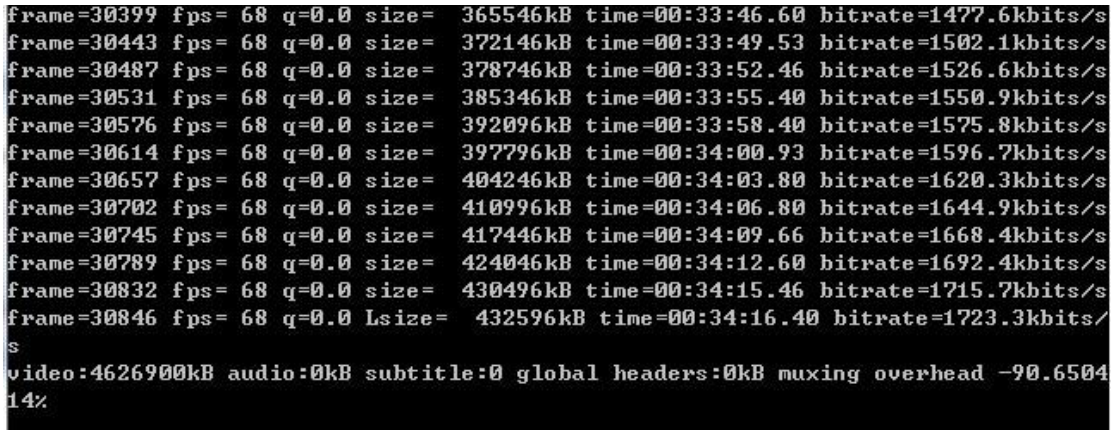



***Fig 3.3.*** *The running environment of ffmpeg(2)*

### 3.1.2. JM Software Introduction

JM is shortcut for Joint Model H.264/MPEG-4 AVC Reference Software, which is developed by JVT(Joint Video Team) of ISO/IEC MPEG & ITU-T VCEG. It is based on the H.264/AVC video compression standard, which is the latest video compression standard. It is mainly used to encode and decode videos. There are many versions ranging from JM 1.0 to JM 18.3 and the latest released one is JM 18.4. In our experiments JM 15.0 was used since it support packet loss, while JM 18.4 does not support this function. So we are going to introduce how to use JM 15.0 instead of JM 18.4.

First basic configuration should be done like described below. The software package contains a Visual Studio .NET workspace named "`jm_vc7.sln`" for .NET 2003 (v7) and another workspace named "`jm_vc8.sln`" for .NET 2005 (v8). In addition, it also contains other workspace for Visual Studio 6 and other working files for UNIX and gcc in windows, which are not discussed here. This workspaces include the following three projects:

- lencod: H.264/AVC reference encoder;
- ldecod: H.264/AVC reference decoder;
- rtpdump: a tool for analyzing contents of RTP packets.

In our experiments, "lencod.exe" and "ldecod.exe" are created in the "bin" directory by compiling "lencod" and "ldecod" project respectively. They are used to encode and decode the videos. The encoder syntax is :

```
Lencod  [-h] [-d defenc.cfg] {[-f curenc1.cfg] [-f curenc2.cfg]...[-f
curencM.cfg]} {[-p EncParam1=EncValue1][-p EncParam2=EncValue2]...[-p
EncParamN=EncValueN]}
```

- `-h` : Prints parameter usage.
- `-d` : Use <defenc.cfg> as default file for parameter initializations. If not used then file defaults to "encoder.cfg"in local directory.
- `-f` : Read <curencM.cfg> for resetting selected encoder parameters. Multiple files could be used that set different parameters.

- `-p` : Set parameter <EncParamM> to <EncValueM>. The entry for <EncParamM> is case insensitive.

The support formats are :

RAW : .yuv , .rgb :

- YUV 4:0:0
- YUV 4:2:0
- YUV 4:2:2
- YUV 4:4:4
- RGB

Here is the command used in our experiments:

```
lencod.exe -d encoder_baseline.cfg -p InputFile="foreman.yuv" -p
OutputFile="foreman_5.264" -p ReconFile="test_rec_5.yuv" -p QPISlice=5
-p QPPSlice=5  -p NumberReferenceFrames = 1 ' '-p OutFileMode= 1
```

where “encoder_baseline.cfg ”is the configure file. The input file is “forman.yuv” and the output file is “forman_5.264”, reconstruction YUV file is “test_rec_5.yuv”. In addition, quantization parameter for intra slices is 5, and quantization parameter for all P slices is 5, too. The number of previous frames used for inter motion search is 1; the output file mode is RTP (Real-time Transport Protocol). When running the encoder , the encoder will display on screen/distortion statistics for every frame. Cumulative results will also be presented. The output information generated may look as following figures which depend on the encoding parameters [10].

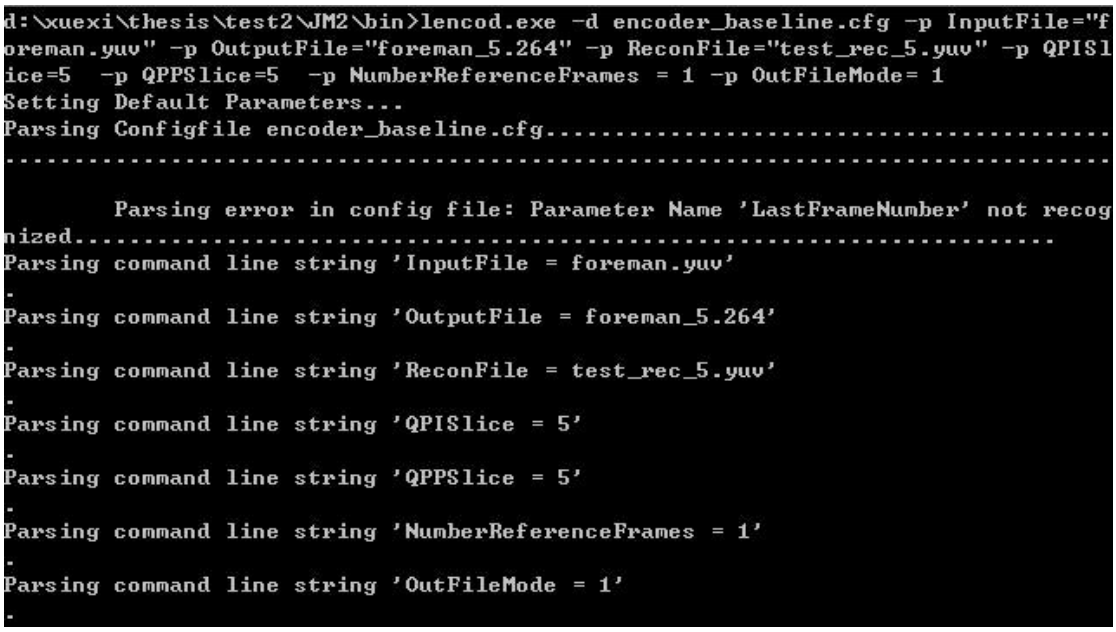
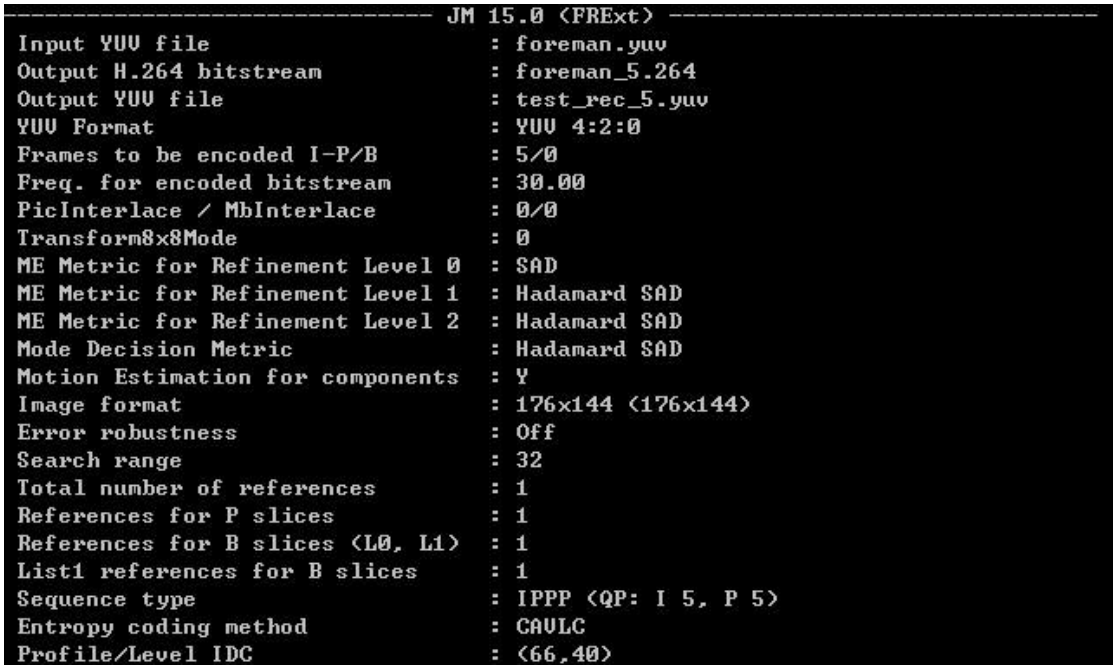
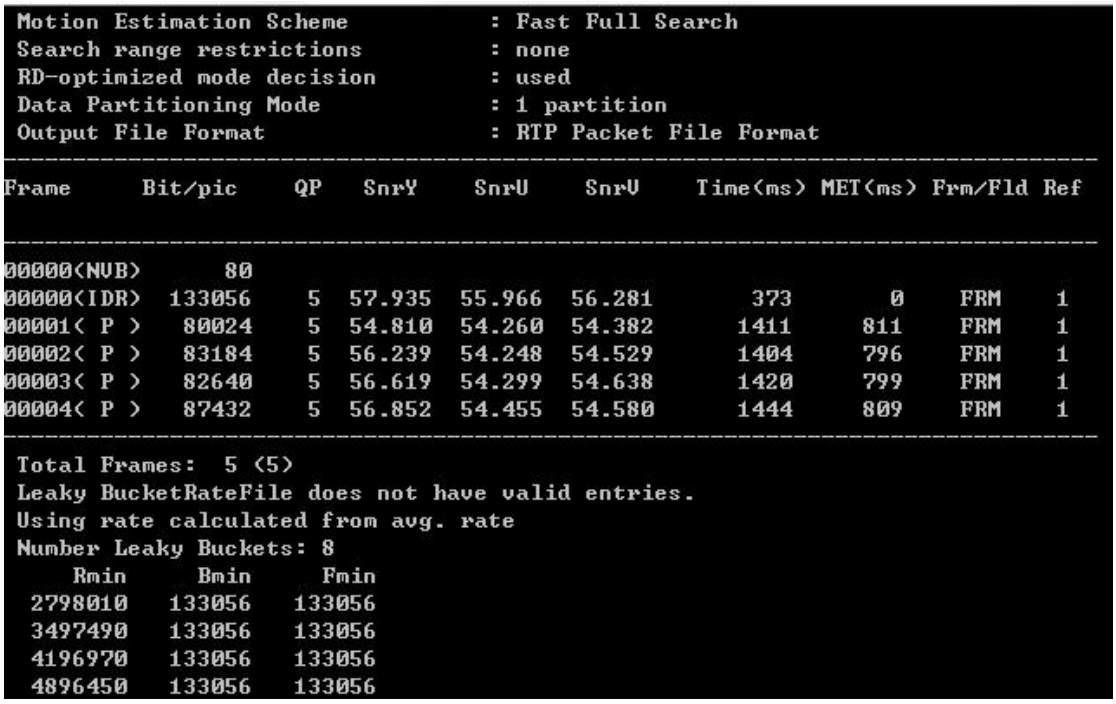



***Fig 3.4.*** *The encoder running displaying(1)*



```
------------------------------ JM 15.0 (FREXt) ------------------------------
 Input YUV file                    : foreman.yuv
 Output H.264 bitstream            : foreman_5.264
 Output YUV file                   : test_rec_5.yuv
 YUV Format                        : YUV 4:2:0
 Frames to be encoded I-P/B        : 5/0
 Freq. for encoded bitstream       : 30.00
 PicInterlace / MbInterlace        : 0/0
 Transform8x8Mode                  : 0
 ME Metric for Refinement Level 0  : SAD
 ME Metric for Refinement Level 1  : Hadamard SAD
 ME Metric for Refinement Level 2  : Hadamard SAD
 Mode Decision Metric              : Hadamard SAD
 Motion Estimation for components  : Y
 Image format                      : 176x144 (176x144)
 Error robustness                  : Off
 Search range                      : 32
 Total number of references        : 1
 References for P slices           : 1
 References for B slices (L0, L1)  : 1
 List1 references for B slices     : 1
 Sequence type                     : IPPP (QP: I 5, P 5)
 Entropy coding method             : CAVLC
 Profile/Level IDC                 : (66,40)
```

***Fig 3.5.*** *The encoder running displaying(2)*



```
 Motion Estimation Scheme          : Fast Full Search
 Search range restrictions         : none
 RD-optimized mode decision        : used
 Data Partitioning Mode            : 1 partition
 Output File Format                : RTP Packet File Format
-------------------------------------------------------------------------------
Frame     Bit/pic    QP   SnrY    SnrU    SnrV    Time(ms) MET(ms) Frm/Fld Ref

-------------------------------------------------------------------------------
00000(NVB)      80
00000(IDR)  133056     5  57.935  55.966  56.281      373       0    FRM    1
00001( P )   80024     5  54.810  54.260  54.382     1411     811    FRM    1
00002( P )   83184     5  56.239  54.248  54.529     1404     796    FRM    1
00003( P )   82640     5  56.619  54.299  54.638     1420     799    FRM    1
00004( P )   87432     5  56.852  54.455  54.580     1444     809    FRM    1
-------------------------------------------------------------------------------
 Total Frames:  5 (5)
 Leaky BucketRateFile does not have valid entries.
 Using rate calculated from avg. rate
 Number Leaky Buckets: 8
     Rmin      Bmin      Fmin
  2798010    133056    133056
  3497490    133056    133056
  4196970    133056    133056
  4896450    133056    133056
```

***Fig 3.6.*** *The encoder running displaying(3)*

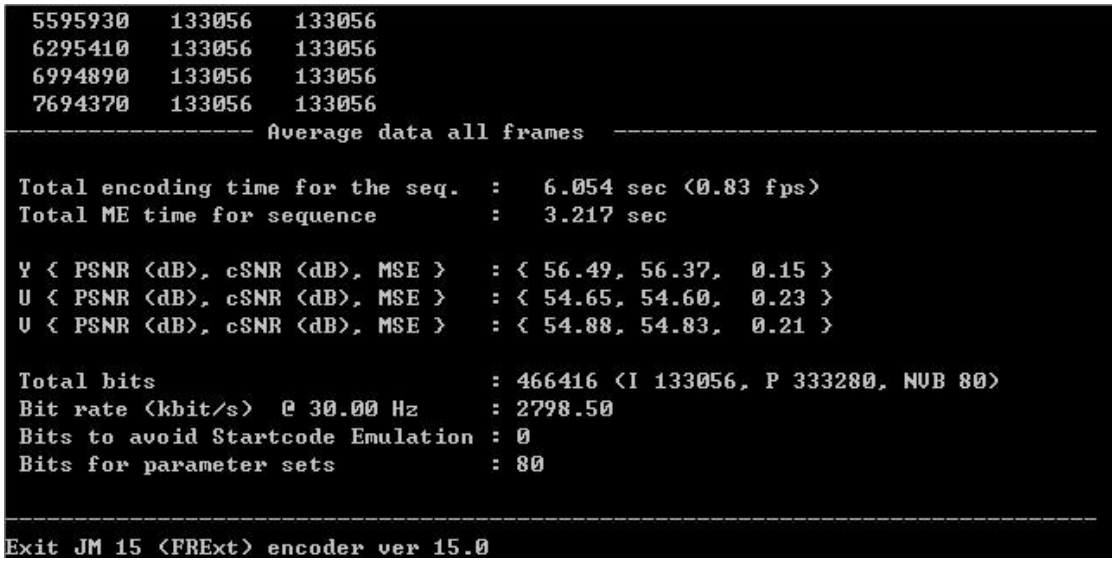



***Fig 3.7.*** *The encoder running displaying(4)*

The decoder syntax is :

```
ldecod [-h] {[defdec.cfg] | {[-p pocScale] [-i bitstream.264]...[-o output.yuv] [-r reference.yuv] [-uv]}}
```

- `-h` : Prints parameter usage.
- `[defdec.cfg]` : Optional decoder config file containing all decoder information
- `-s` : Silent decoding;
- `-i` : Decode file<bitstream.264>. Default is set to “test.264”;
- `-o` : Reconstructed file name is set to <output.yuv>. Default is test_dec.yuv;
- `-r` : Reference sequence file for PSNR computation is set to <reference.yuv>. default is test_rec.yuv;
- `-p` : Set Poc Scale to the value pocScale. Default is 2;
- `-uv` : Output 400 content with gray chroma components, to allow viewing of output on 420 YUV players.

Decoder parameters need to be placed in a specific order for the decoder to work correctly. Some parameters which may need to modify in the experiment are:

- NAL mode (0 = Annex B, 1= RTP packets) should be 1;
- Error Concealment option (0= disabled/default, 1= frame copy, 2=motion copy) should be 1 [10].

And the command used in the experiment is :

```
ldecod.exe decoder.cfg
```

When running the decoder, the decoder will display on screen rate/distortion statistics for every encoded frame. Cumulative results will also be presented. The displaying results is shown below, which is depended on different parameters:

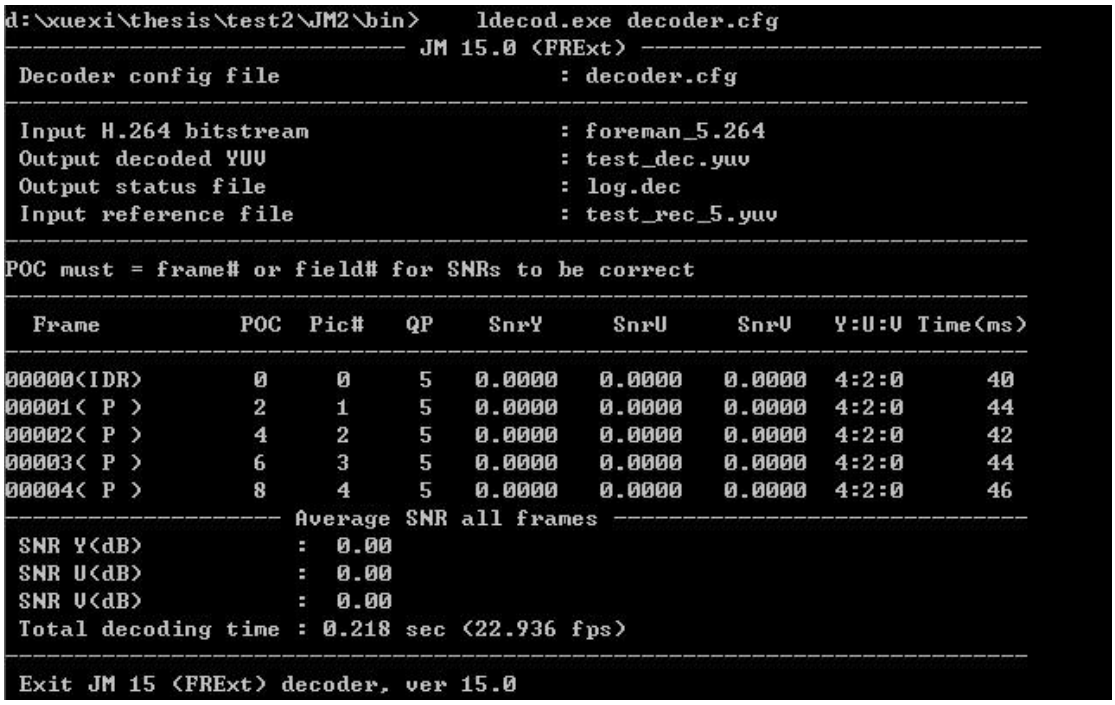



***Fig 3.8.*** *The decoder running displaying*

## 3.2. Packet Loss Procedure

In this section, the packet loss procedure is introduced. There are two different queue mangements to be simulated. Drop tail results in bunches of packet loss, representing the correlated packet loss process while RED leads to completely uncorrelated packet loss process. Thus, we introduce packet losses artificially generating first a sequence of them and then dropping packets in our sample videos. The sequence we would use are the sequence of 0 and 1, where 1 indicates a dropped packet, while 0 indicates the received one. In order to generate sequences with the same packet loss rate but different correlation, discrete-time autoregressive process of order one DAR(1), is used. So, in what follows we describe the idea of DAR(1) and the process to inject packet loss to the encoded sequence.

### 3.2.1. Generating Random Sequence

To generate a random sequence of ones and zeros, many random process models can be chosen. The reason why DAR(1) is selected is that it is easy to implement. Another reason is that DAR(1) can simulate both Drop tail and RED just by modifying the two parameters which represent the packet loss probability and correlation respectively and it is widely used in packet traffic modeling nowadays. In addition, DAR(1) is thought to be one of the most tractable and convenient models for practical use.

DAR(1) is defined as fellows. It is assumed that the time axis is divided into chips of a fixed length. Let's $B_n$'s ( $n = 0,1,2,...$ ) denote independent and identically distributed

(i.i.d) random variables that take a value in $N=\{0,1,2...\}$. Let $B$ denote a generic random variable of the $B_n$'s ($n=0,1,2,...$), which follow a distribution $\Pr[B=m]=b_m(m\in N)$, Let $\alpha_n$ 's( $n=0,1,2,...$ ) denote i.i.d. Random variables with $\Pr[\alpha_n=1]=p$ and $\Pr[\alpha_n=0]=1-p$ .It is assumed that the $\alpha_n$'s are independent of the $B_n$'s. Let $A_n(n=0,1,2...)$ denote the batch size. The $A_n(n=0,1,2...)$ is then determined below.

$$\begin{cases} A_0=B_0 \\ A_n=\alpha_n A_{n-1}+(1-\alpha_n)B_n, \quad n=1,2,..., \end{cases}$$

or equivalently,

$$\begin{cases} A_0=B_0 \\ A_n=\begin{cases} A_{n-1}, & c=p \\ B_n, & c=1-p \end{cases} \quad n=1,2,..., \end{cases}$$

It is known that $A_n$ which is marginal distribution is identical with the distribution of $B$. Allowing the sequence $B_n$ ( $n=0,1,2,...$ ) only take two values 1 and 0 with the probabilities $p_L$ and $1-p_L$. The binary DAR(1) process is obtained [9,13].

### 3.2.2. Introducing Packet Losses

According to previous section, the model in our experiments is a binary DAR(1) model. The chosen parameters are as follows:

$$B_n=\begin{cases} 0, & p=p_L \\ 1, & p=1-p_L \end{cases} \quad p_L=0.001,0.002,...0.010, \quad n=0,1,2,....,$$

$$\begin{cases} A_0=B_0 \\ A_n=\begin{cases} A_{n-1}, & c=p_c \\ B_n, & c=p_c \end{cases} \quad p_c=0.1,0.2,...0.9 \quad n=0,1,2,..., \end{cases}$$

In the formulas above, $p$ is packet loss probability and $c$ is correlation. In addition, the sequence $A_n(n=0,1,2,...)$ is the "0"and "1" sequence that we want [9].

After deciding the suitable model, we begin transfer the model into a C programming file. In this file time is used as a seed of function `srand()`. The generated random sequence is recorded in a file named "generatedDAR.txt". Because the first three frames carry synchronization an version information that must be delivered to the destination for decoding to strat,, the first three packets should not be dropped, otherwise , the decoder would give error.

After encoding the video and generating the random sequence with certain packet loss probability and correlation, packet loss is injected to the encoded video. Here is the code injecting packet losses:

```
load generatedDAR.txt;
% Injecting packet loss to the the encoded sequence and then decoding again
    fid = fopen(['foreman_5.264']);
    fido = fopen(['foreman_err_5.264'], 'w');
    pNum=0;
    i=1;
    t=0;
   % while (~feof(fid)) %shoufoudaowenjianwei
        %new packet
        pSize=fread(fid,1,'*uint32')
        if (feof(fid))
             break
         end
        pTime=fread(fid,1,'*uint32');
        data=fread(fid,pSize,'*uint8')
        pNum=pNum+1
        c= generatedDAR(i)
        t=t+1
        if (generatedDAR(i) == 0)
            6
           fwrite(fido, pSize , 'uint32');
           fwrite(fido, pTime , 'uint32');
           fwrite(fido, data , 'uint8');
        end
         i=i+1;
    %end
fclose (fid);
fclose (fido);
```

***Fig 3.9.*** *Packet loss coding*

The idea of injecting packet losses to encoded video is to copy the frames whose loss indicator equals to 0 from the encoded ".264" file to a new ".264" file. The frames whose loss indicator equals to 1 are "dropped". The new ".264" file is the distorted encoded video.

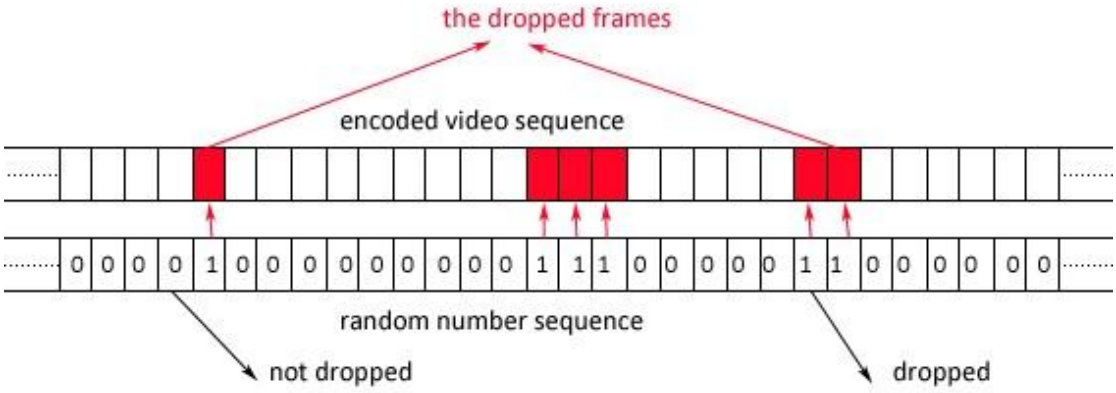



***Fig 3.10.*** *The interpretation of injecting packet loss*

After the distorted encoded video is generated, decoding procedure begins in order to obtain the distorted output video. The command of decoding is shown below:

```
ldecod.exe decoder.cfg
```

## 3.3. Video Quality Evaluation

So far, the simulation of video transmission is done, that is, the distorted video is obtained. Now we need to estimate the quality of video, using an onjective QoE metric. In our experiments, PSNR is chosen as the metric for QoE, since it is easy to implement and it gives easily interpretable results compared to SSIM. PSNR is calculated according the grey values of frames, which are most sensitive to packet losses. Recall that SSIM is less sensitive to packet loss. The values of PSNR of each frame between distorted and original video are calculated and the PSNR of the video is the average value of PSNRs of all these frame.

The calculation of PSNR requires the same dimensions of frame metrics between original video and distorted one. During the process of injecting packet losses to the encoded video, when the packet loss probability and correlation are both very high, for example $(c, p) = (0.9, 0.007)$, some frames can be totally lost. Alternatively, it could be the case that a part of frame is lost. The loss of a whole frame causes some empty cells in the video metrics, which lead to the situation that the dimensions of frame metrics between original video and distorted one are different. That makes impossible to calculate PSNR using the definition introduced in the previous Chapter. The reason of this problem is that JM 15.0 fails to provide a effective method to complement or mend the loss or distorted parts of video, especially, when the packet loss is high or distortion is very serious. So, in order to solve the problem, the empty cells are replaced by the cells whose elements are all set to 0. After adding these cells into the video metric, when displaying the video, the lost parts become black frames, which is a little bit different from practical situation. However, this approach is not only convenient but also still acceptable and practical. Although, it leads to lower PSNR values in the results which would be discussed in the next chapter, it gives better QoE to audience.

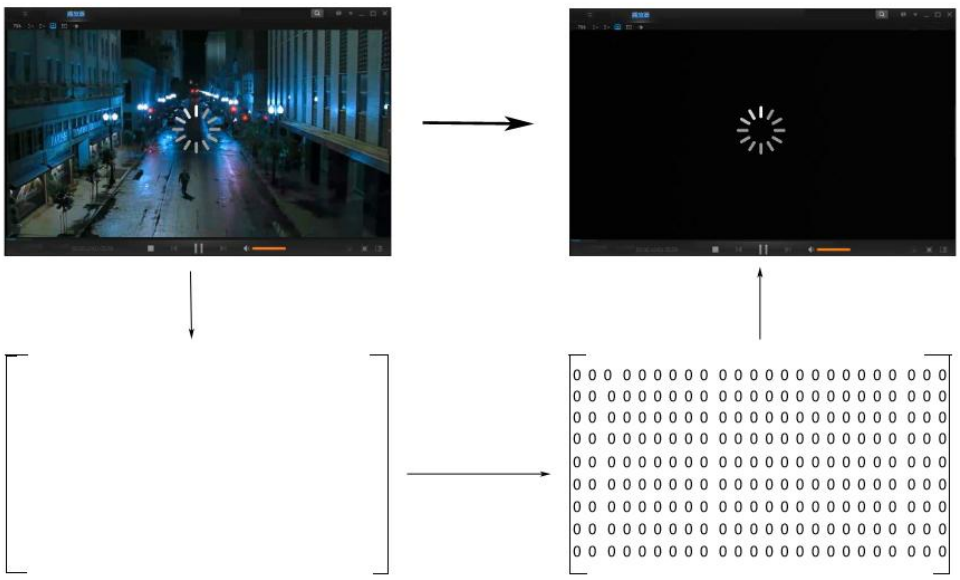


***Fig 3.11.*** *The interpretation of add 0 in to empty cell*

## 3.4. Supplement

In order to provide reasonable results , the input parameters (*p*,*c*) are selected as shown in Table 3.1. Since it is a statistical experiment, we do the same procedure to all the selected videos. Sometimes, some sample points need to be estimated several times.

***Table 3.1.****The table of sample points*

| c (correlation ) | p (packet loss probability) |
|---|---|
| 0.0 | 0.001 |
| | 0.003 |
| | 0.005 |
| | 0.007 |
| | 0.010 |
| 0.4 | 0.001 |
| | 0.003 |
| | 0.005 |
| | 0.007 |
| | 0.010 |
| 0.9 | 0.001 |
| | 0.003 |
| | 0.005 |
| | 0.007 |
| | 0.010 |

# 4. NUMERICAL RESULTS AND ANALYSIS

The chapter is consist of three parts. In the first part the results of the experiments will be given and discussed. In second part, the summary of the relationship between network-intrinsic metrics and QoE of video streaming, will by discussed. In addition, the further opinions behind the results would be discussed in the last part. The final conclusions would be discussed in next chapter.

## 4.1. Experiment Result

In this section, the results are presented in terms of graphs and tables. The interpretation of the results would be also given.

Here are the tables of results. The PSNR values of all the input parameters in the following table are the averaged values which excludes the abnormal values (outliers) which cannot be avoided in statistical experiments.

***Table 4.1.*** *"short.yuv" the numerical results(1)*

| **Short: lasts for around 32 minutes and 30846 frames in total** | | | |
|---|---|---|---|
| **c** | **p** | **psnr** | **ssim** |
| 0 | 0.001 | 33.2056 | 0.9965 |
| | 0.005 | 32.6571 | 0.9965 |
| | 0.007 | 32.1891 | 0.9965 |
| | 0.01 | 31.8683 | 0.9965 |
| 0.4 | 0.001 | 33.2032 | 0.9965 |
| | 0.003 | 21.8499 | 0.000003 |
| | 0.005 | 12.9904 | 0.5538 |
| | 0.007 | 12.9902 | 0.5538 |
| | 0.01 | 12.6534 | 0.5538 |
| 0.9 | 0.001 | 31.4451 | 0.5538 |
| | 0.003 | 19.3403 | 0.5538 |
| | 0.005 | 14.8884 | 0.5538 |
| | 0.007 | 13.0548 | 0.5538 |
| | 0.01 | 18.9559 | 0.5538 |

***Table 4.2.*** *"medium.yuv" the numerical results(2)*

| **Medium: lasts for around 36 minutes and 36815 frames in total** | | | |
|---|---|---|---|
| **c** | **p** | **psnr** | **ssim** |
| 0 | 0.001 | 33.0357 | 0.9959 |
| | 0.003 | 32.8134 | 0.9959 |
| | 0.005 | 32.6119 | 0.9959 |
| | 0.007 | 32.5553 | 0.9959 |
| | 0.01 | 3.8404 | 0.0003 |

| | | | |
|---|---|---|---|
| 0.4 | 0.001 | 33.0581 | 0.9959 |
| | 0.003 | 24.5469 | 0.4558 |
| | 0.005 | 17.9676 | 0.4558 |
| | 0.007 | 13.671 | 0.4558 |
| | 0.01 | 12.7023 | 0.4558 |
| 0.9 | 0.001 | 31.309 | 0.4558 |
| | 0.003 | 26.8872 | 0.4558 |
| | 0.005 | 11.7074 | 0.4558 |
| | 0.007 | 11.8693 | 0.4558 |
| | 0.01 | 14.5765 | 0.4558 |

***Table 4.3.*** *"long.yuv" the numerical results(3)*

| **Long: lasts for around 42 minutes and 45534 frames in total** | | | |
|---|---|---|---|
| **c** | **p** | **psnr** | **ssim** |
| 0 | 0.001 | 33.6018 | 0.9966 |
| | 0.003 | 33.807 | 0.9966 |
| | 0.005 | 32.6583 | 0.9966 |
| | 0.007 | 32.2392 | 0.000003 |
| | 0.01 | 20.2294 | 0.0003 |
| 0.4 | 0.001 | 27.2021 | 0.3483 |
| | 0.003 | 14.1343 | 0.3483 |
| | 0.005 | 13.3739 | 0.3483 |
| | 0.007 | 13.629 | 0.3483 |
| | 0.01 | 14.9208 | 0.3483 |
| 0.9 | 0.001 | 33.8942 | 0.3483 |
| | 0.003 | 21.9849 | 0.3483 |
| | 0.005 | 13.3226 | 0.3483 |
| | 0.007 | 12.272 | 0.3483 |
| | 0.01 | 17.1089 | 0.3483 |

Next, some graphs are generated by the data above in order to get the relation between packet loss probability ( $p$ ), correlation ( $c$ ) and PSNR, that is, showing $psnr = f(p,c)$. It is just a reference results from the experiment. The sample points of PSNR values in the graphs below are all the averaged values of several experiments. They also exclude the abnormal values (outliers). It should be mentioned that we are dealing with statistical experiments and the results are limited by the number of experiments. The author cannot perform infinitely many of them because of the limitation due to some objective reasons like project time, computer access and so on. Due to this the results may be not that precise. However, it still reasonable and much acceptable and it still shows the relationship between packet loss probability( $p$ ), correlation( $c$ ) and PSNR expressing $psnr = f(p,c)$.

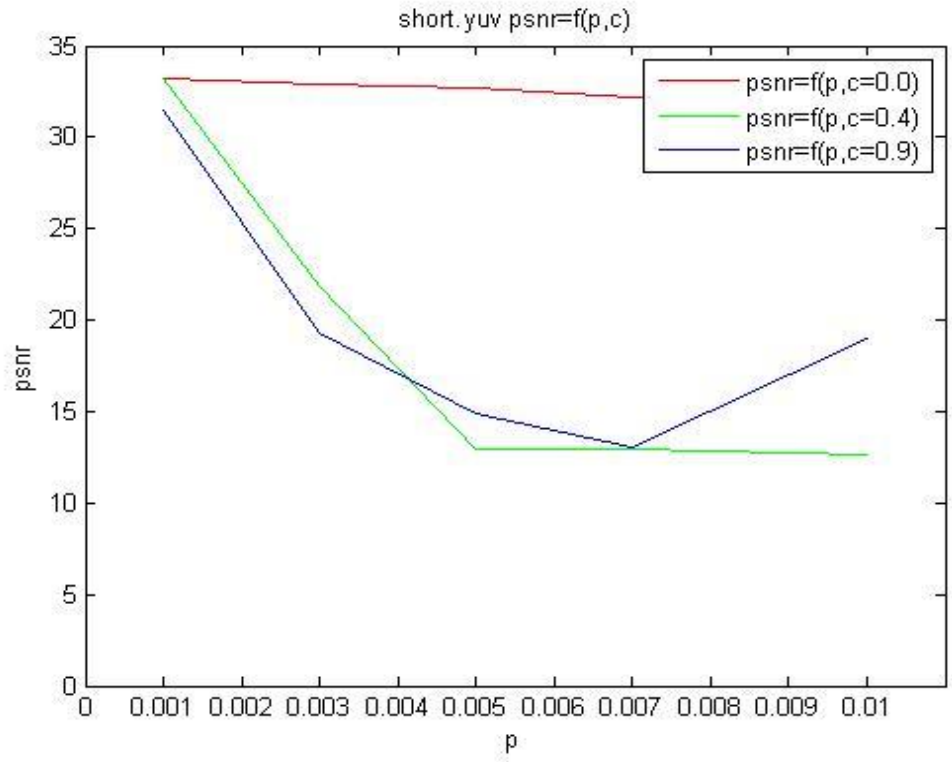



***Fig 4.1.*** *The relation between (p,c)and psnr (1).*

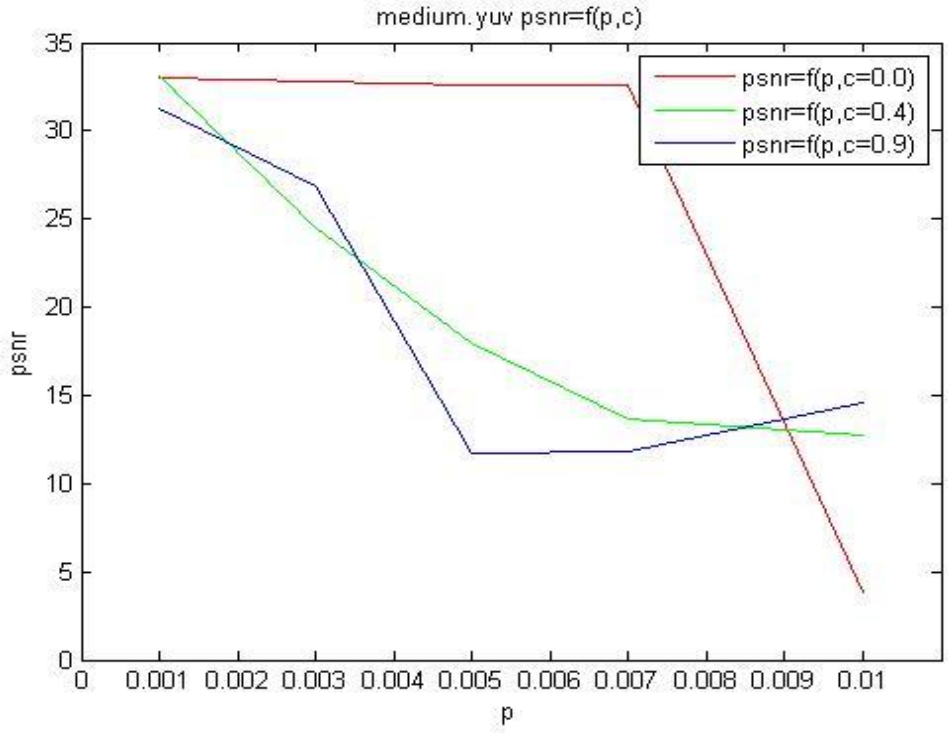



***Fig 4.2.*** *The relation between (p,c)and psnr (2).*

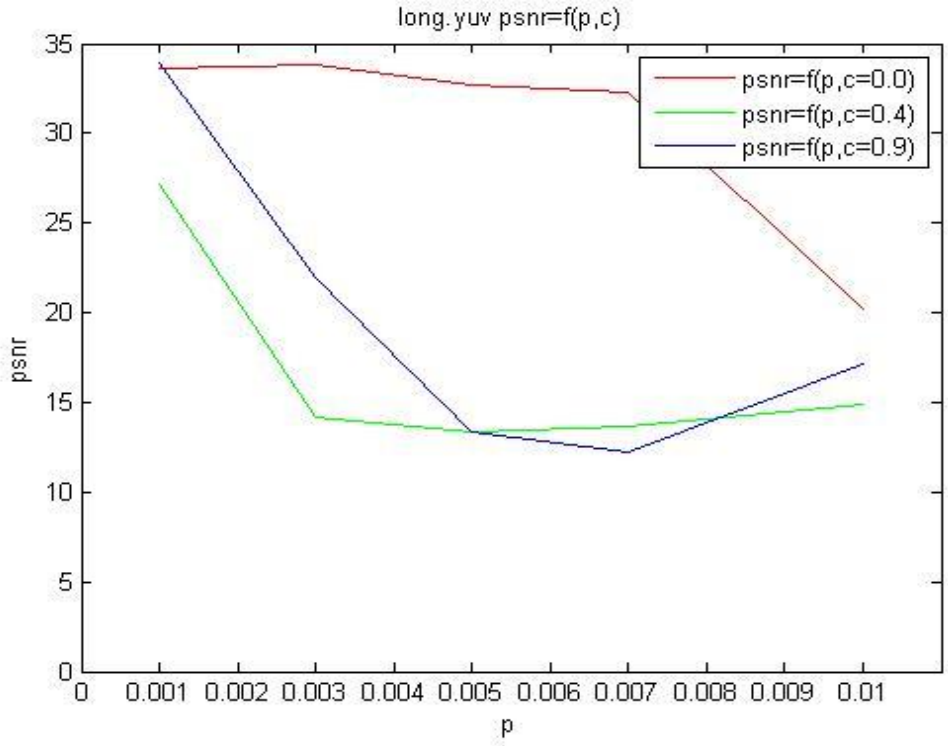



***Fig 4.3.*** *The relation between (p,c)and psnr (3).*

From the graphs, we can see that for a certain correlation value, for example, $c = 0.0$, the higher packet loss probability is, the lower PSNR is. Secondly, there are always key points where the PSNR changes quickly (abruptly), for example $(c = 0.0, p = 0.007)$ and $(c = 0.9, p = 0.005)$. The relationship between the packet loss probability and PSNR for a certain value of correlation, $psnr = f(p,c)$, is not linear. There are huge differences between the lines with different values of correlation. When correlation is very low , $c = 0.0$ or $c = 0.1$, and the probability of packet loss is very high like $p > 0.007$, PSNR values decreases quickly and sharply, that is to say, in this case, QoE of the video is impacted greatly. While , when correlation is higher, say, $c > 0.4$, and the probability of packet loss is not that high, like $p = 0.005$, PSNR values becomes very low, falling down even below 20. So, we can see that correlation of packet loss process have a great impact on QoE.

In next section, a further opinions and summary about the relationship between network -intrinsic metrics and QoE is given according to this experimental results.

## 4.2. New Network-intrinsic QoE Metrics

After analyzing the experiment results, some basic idea about the relationship between packet loss probability and correlation and PSNR or QoE is obtained. From the previous results, we can infer some further ideas of the relationship between packet loss probability and correlation and PSNR, and the detailed situations in different cases. The impact of different correlation can be in four categories, depending on whether the video suffers drastic effect or not. In this section, these different cases are discussed.

First, in the situation when correlation is very low, i.e. $c = 0.0$, $c = 0.1$ or $c = 0.2$, there is no drastic impact on the video sequence. That is to say, even though, there are some lost packets, they are almost all well separated and the interval between impacted frames is very big. The lost content of the video is very little and only few parts are in fact impacted. Since the lost content is totally separated it fails to be found by the audience in subjective tests. So, in this situation, video transmission get the best possible conditions. If the packet loss probability is very low, QoE would be very good, otherwise, QoE is not good but it is still acceptable to the audience . The situation in this case is as shown in Fig. 4.4.

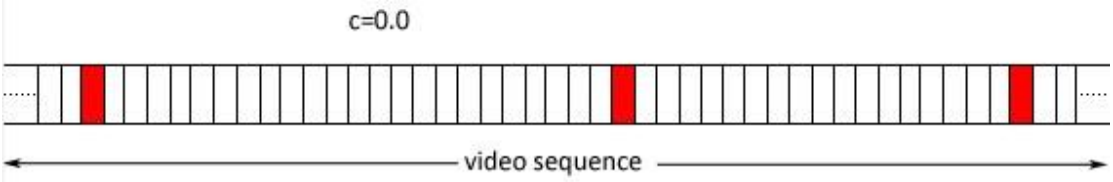



***Fig 4.4.*** *The impact when correlation is very low.*

The second case we consider is when correlation is very high like $c = 0.8$, $c = 0.9$ or $c = 0.99$. In this case there is a drastic impact on one segment of the video. All the lost packets are organized in a batch. There are many frames like 3000 frames in this segment, and the exact number of impacted frames depends on the situation of the network and the exact packet loss probability. Since these impacted frames are continuous and the number is very huge, the lost or distorted content of the video is very much and, thus, can be detected easily by audience and may lasts for a long time. The procedure of watching is seriously disturbed. However, it happens only once during the transmission of the video. So, in this situation, QoE is affected by the correlation greatly, but it is still not the worst possible situation. In principle, the distorted parted can be tolerated as the packet loss probability has less effect on QoE. Even if the packet loss probability is very low, QoE is still not satisfactory. The situation in this case is as shown in Fig. 4.5.

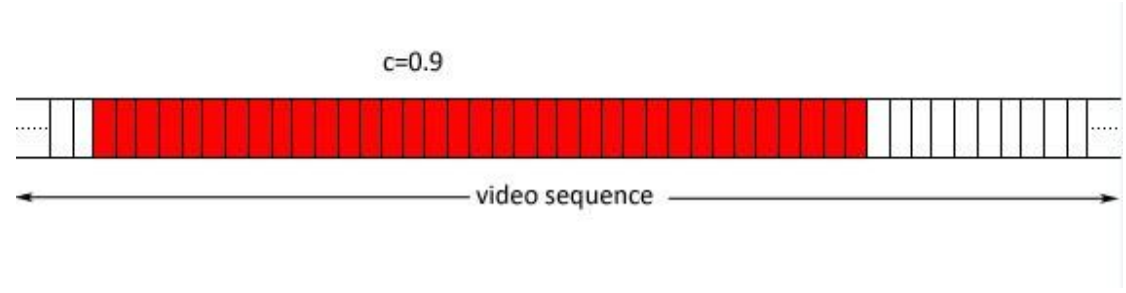



***Fig 4.5.*** *The impact when correlation is very high.*

The third case is when the correlation is around 0.5. In this case, the lost packets are divided into several segments and in each segment, the lost packets are continuous. Each segment lasts relatively short time. The number of impacted frames are less than

that of the second situation, but still much more than that of the first situation. For one segment, there are continuously lost packets and the number of them depends on the packet loss probability. When the packet loss probability is very low like $p = 0.1,0.2$, QoE is still acceptable, and there are several small time slots in which the content is lost It may produce only little effect on the displayed picture and PSNR. However, as the packet loss probability increases, the lost content in one segment becomes more and more drastic, and it can be detected by the audience performing a subjective test. People have to endure the content loss or distorted parts several times during watching the video, which gives the worst user experience to the audience. So, in this case, we have the worst possible perceived quality. The situation in this case is illustrated in Fig. 4.6.

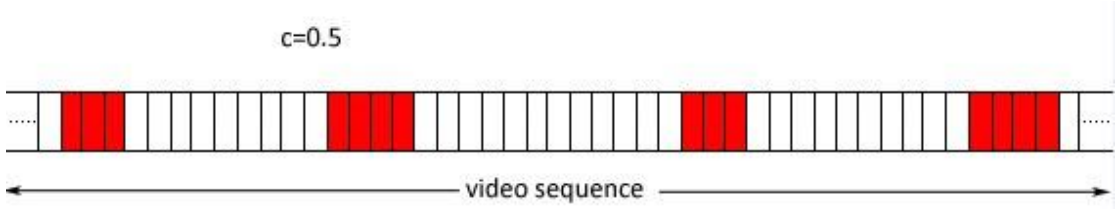



***Fig 4.6.*** *The impact when correlation is around 0.5.*

Last, there is an extreme situation, which may or may not happen in reality. In this case, correlation is extremely huge, i.e. 0.999999...999999999 approaching 1. All the lost packets are organized in a single segment. It may last very long time, and the exact time cannot be reliably predicted . In this case, the video is often given up by the audience as it is totally unwatchable. In this case we cannot reliably assess the video quality. This situation is illustrated in Fig. 4.7.


***Fig 4.7.*** *The impact when correlation is approach to 0.5.*

The obtained results is pretty interesting. They are quite different from what we expected to get before the experiments. At the beginning, we thought that the higher correlation may produce lower PSNR and the relationship would be linear. But the result is totally different which can be seen from the analysis above. These results are more close to the results of subjective method. In order to explore it, the author also did a small test to study the results. The author found ten people to see the distorted videos with different correlation but a certain packet loss probability and let them give their estimations of the perceived quality. Eight people gave the lowest estimation to the distorted video whose correlation is 0.5, while only two people gave the lowest estimation to the distorted video whose correlation is 0.9. Notice that this is just a small

test to compare the the obtained objective results with the result of subjective method. If a more precise result is wanted, the number of persons should increase.

Next, we are going to sum and infer the graphs and analysis above in order to get the new network -intrinsic QoE metrics $psnr = f(p,c)$. The summary graph of a metric is shown in Fig. 4.8.

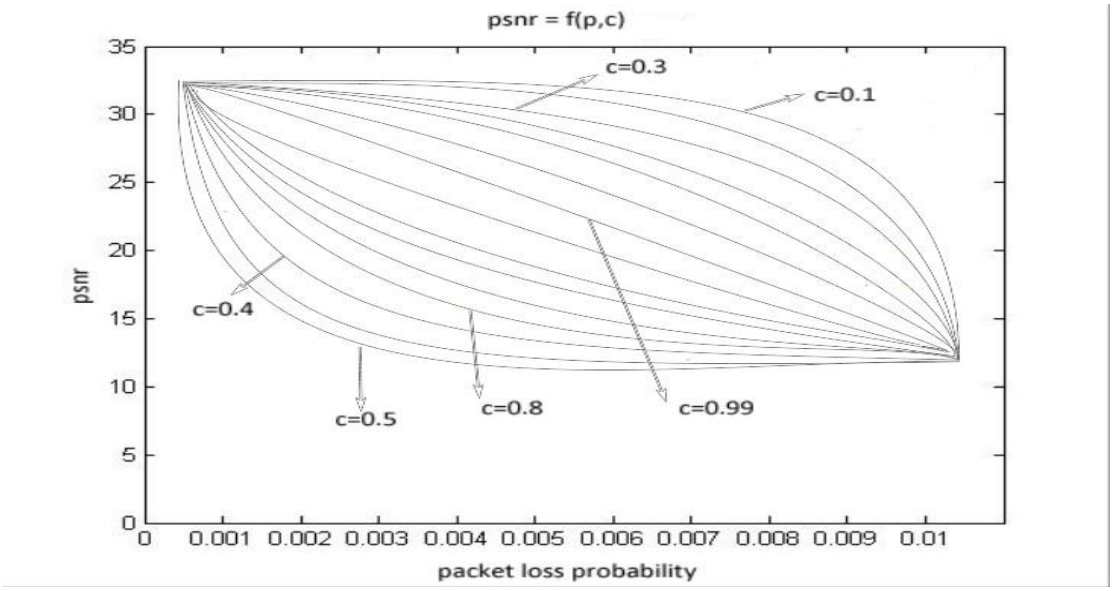



***Fig 4.8.*** *New network-intrinsic QoE metrics.*

It is an reference model, and the values in the graph are all reference correlation values, which are used to represent the point of great changes in PSNR. It is a draft of the metric and the more exact values and associated dependencies can be obtained performing more statistical experiments. However, notice that the exact values are not extremely essential as, in fact, the metrics still describe the relationship. In this graph, when correlation is very low, around 0.0, 0.1 or 0.2, the QoE is satisfactory when the packet loss probability is not very high. As correlation increases, QoE suffers a drastic effect, around the point with c=0.5. The change is huge as PSNR drops exponentially fast. Even the packet loss probability is quite low, QoE is still not good. Next, when the correlation increases again , the PSNR values become higher again comparing to the previous case when the correlation was around 0.5. In this case, QoE still decreases but it is acceptable to the audience. So, we can see that the correlation has a great impact on PSNR instead of the conventional opinion that the packet loss probability is the only important networking factor affecting the percived quality.

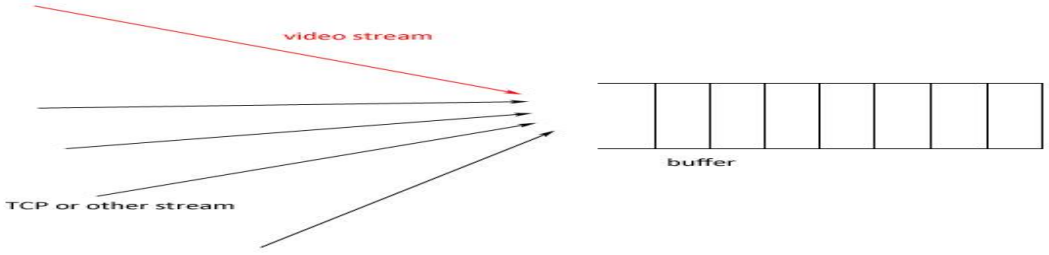



***Fig 4.9.*** *Different streaming to a router.*

Let us get back to the queuing disciplines we studies in this thesis. As we know, a router buffer would accept different kinds of packet streams which may include video streaming , audio streaming, and TCP packets. The queue management is often drop tail whose operationhas been explained in the previous chapter. Reaching the full capacity of a buffer indicated that the congestion happened in the network. At this time the router begins to drop all the arrival packets no matter which stream they belong to, video streaming or TCP. Video streaming begins to lose packets continuously, and the exact number of lost packet is decided by the detailed state of the network. At this time, TCP starts its congestion control mechanism. The aggregated arrival stream of packets becomes thinner. When the buffer again have space to accept the arrival packets, TCP packets begins to be transmitted again, and most TCP sources in this time are in the slow start regime. Congestion may happen again, and video streaming may starts losing packets continuously again. Thus, congestion may happen many times and each congestion may lasts a relatively long time. The video may experience distortion many times, and this situation is similar to the cases analyzed in this thesis, where correlation is around 0.5.

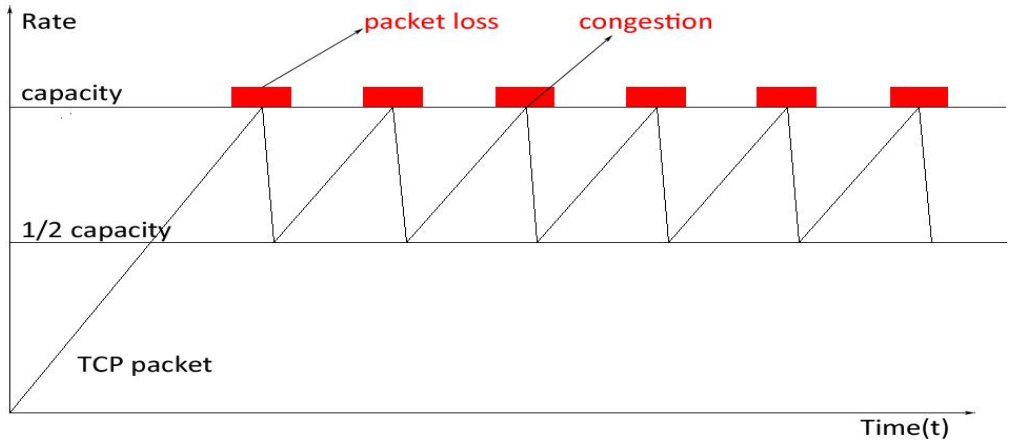



***Fig 4.10.*** *Network situation.*

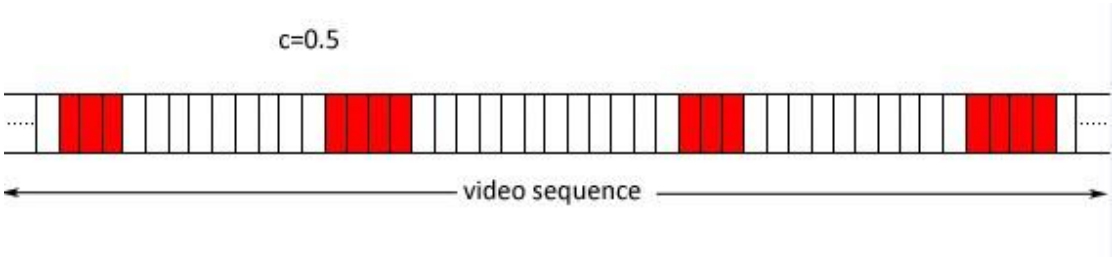
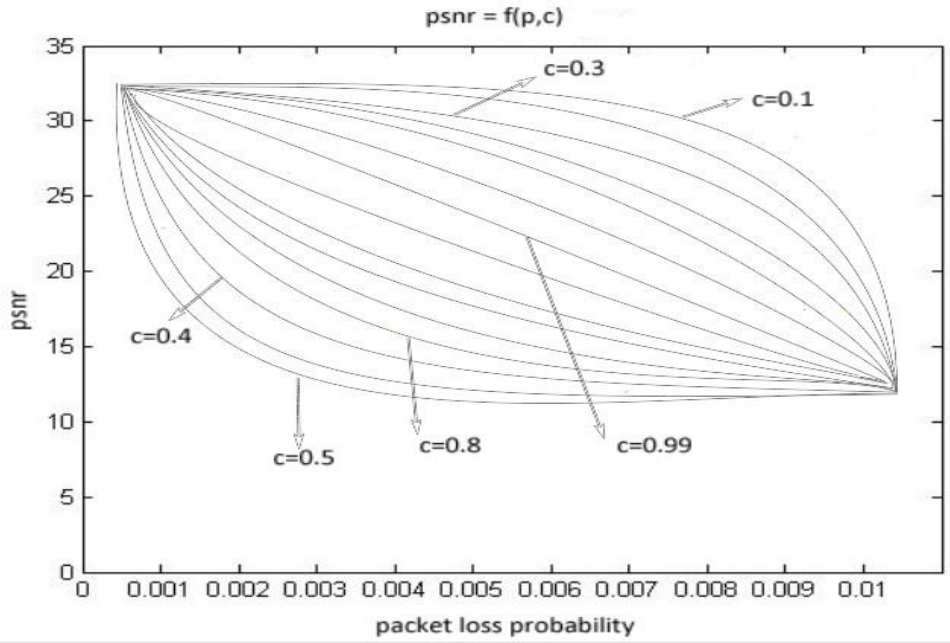



***Fig4.11.*** *Video situation.*

If the congestion happens just once but lasts a very long time it may resemble the situation with c=0.9. The exact values of packet loss possibility and correlation are very hard to estimate and they are not of great important here. Since the network situation is highly dynamic and complicated, we can hardly be sure what kind of detailed situations may happen. The interpretation above is just one of possible situation , but it is still very convincing.

# 5. CONCLUSION

After discussing some theoretical knowledge and analyzing the numerical results, we have already known the idea of video streaming and compression techniques; video evaluation; router queue disciplines and conventional network intrinsic metrics; the experiment flow and the relationship between network intrinsic metrics and QoE. In this chapter we provide our conclusions. The conclusions are divided into two parts: one is about the new metrics and another is about queue management.




***F**igure 5.1. Network intrinsic QoE metrics*

First, we talk about the new network intrinsic QoE metrics for video transmission. It provides the relationship between network intrinsic metrics (packet loss probability and correlation) and QoE (PSNR in this paper): $psnr = f(p,c)$. Even though it is just a reference model, we clearly demonstrated that the correlation have a great impact to QoE and should be taken into account when estimated perceived quality of video. Recall that the conventional method to estimate QoE is to compare the distorted video and original one. It is impossible to implement in networking environment, since it is hard for end user to get the original video. Our metrics can estimate the video quality in a different way. We can use the network parameters to estimate QoE of video instead of using the the comparison between the original video and the distorted one. This approach is easy to implement. The matric is useful for service providers. Indeed, the video website can modify their codec standards according to the network situation and this decision can be based on our metric. Moreover, our metric can help network

operators to know the situation of network. It is  way easier to find the fundamental reasons which can lead to the bad network situation suing the proposed metric.

Second thing is related to the router queue management discipline. Drop tail is easy to implement, and it requires no configuration or maintenance. The performance of drop tail is not bad so far, however, it sometimes allows severe congestions to happens disturbing the network operation for a (relatively) long periods of time. Most network operators still use this queue management just because of its simplicity. However, as the volume of data is increasing sharply, especially, in wireless networks the congestions will happen more often. It brings continuously lost packets to video and HD movies that are very sensitive to the distorted frames. So, drop tail may not be a good fit for the Internet in the near future. Easy configuration and maintenance is not that important comparing to the demand of higher QoE from audience.
Are there some better managements to avoid congestion in the network? I can say RED is a better solution to substitute drop tail. RED can drop arrival packets randomly according to the probability *p*, and may prevent packets from being dropped continuously effectively. Because of this, RED is a good alternative for video streaming, especially, for HD video. From the previous analysis, if distorted or lost video content is little and distortions are well separated in time, the perceived video quality is affected only little which may not be noticeably to the audience . Usually, if the audience fails to detect the distorted part, it can be ignored and it is thought to be no impact to video quality. I have to say RED is better than drop tail on dropping packets. The worst case is when RED drops all the arrival packets, which is the same as drop tail. However, it happen rarely and requires very special network conditions. The worst case happens more often when using drop tail.
In addition, we need to mention that RED is also a good fit for TCP. A short interpretaation is still given here in order to explian the advantages of RED , even though it  has little relationship between the main topic.completely As we can see in Fig. 5.2 and Fig. 5.3, the red part is the used bandwidth, the used bandwidth of RED is much smaller than that of drop tail. From previous statement, we can see that RED is good for not only video streaming, but also TCP. It can handle the huge data effectively. In addition, nowadays most routers have this management, its configuration is not very difficult,which need to configure two more threshold values. So RED is highly recommended to practice.

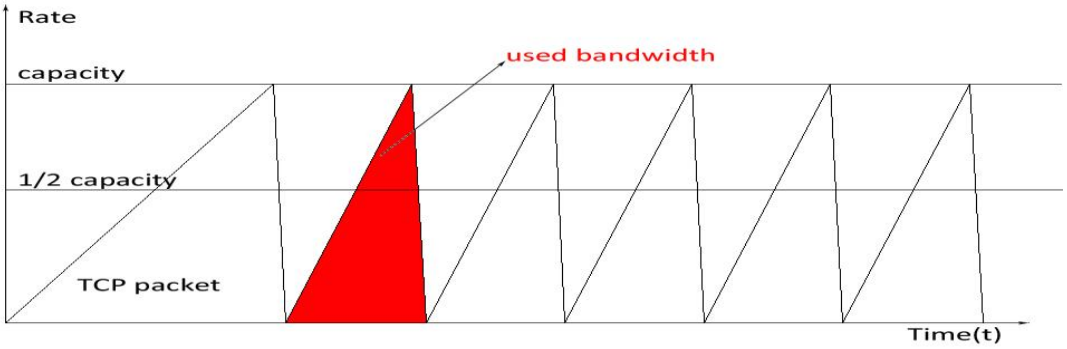



***Fig 5.2.*** *Used Bandwidth for drop tail*

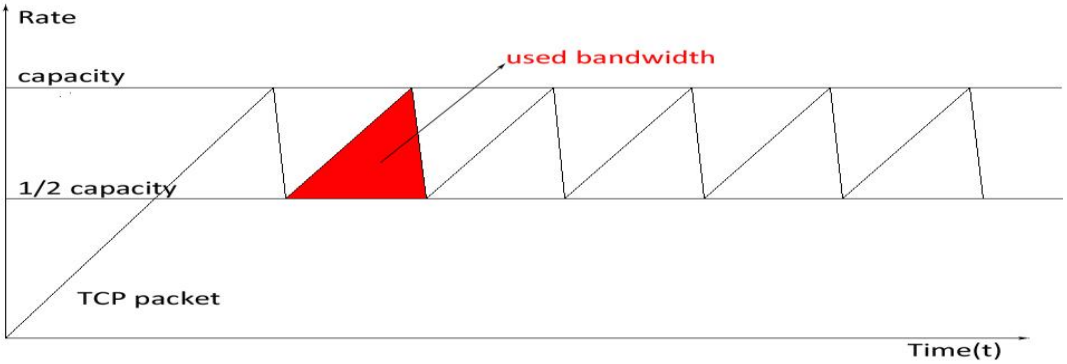



***Fig 5.3.*** *Used Bandwidth for RED*

## REFERENCES

[1.]Djordje Mitrovic . Video Compression[WWW] . Available at : http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/AV0506/s0561282.pdf

[2.]Sanjeev Patel ,Pankaj Gupta,Ghanshyam Singh. Performance measure of Drop tail and RED algorithm. 2010 International Conference on Electronic Computer Technology (ICECT),7-10 May 2010,Kuala Lumpur. Electronic Computer Technology(ICECT)2010 International Conference on,pp. 35-38

[3.]BT.500-11. Methodology for the subjective assessment of the quality of television pictures.ITU-R ,06,2002. 48p

[4.]Tong Yubing, Zhang Qishan, Qi Yunping. Image Quality Assessing by Combining PSNR with SSIM. Journal of Image and Graphics. 11(Dec.2006)12,pp.1758-1763

[5.]Alain Horé , Djemel Ziou. Image quality metrics: PSNR vs. SSIM. 2010 International Conference on Pattern Recognition,23-26 Aug. 2010,Istanbul. Pattern Recognition (ICPR), 2010 20th International Conference on,pp.2366 - 2369

[6.] ITU-T SG12. Definition of Quality of Experience . TD109rev2(PLEN/12),Geneva, Switzerland. Jan 2007,pp.16-25

[7.] Wael Cherif, Adlen Ksentini, Daniel Négru,Mamadou Sidibé. A_PSQA: Efficient real-time video streaming QoE tool in a future media internet context.2011 IEEE International Conference on Multimedia and Expo,11-15 July 2011,Barcelona, Spain.Multimedia and Expo (ICME), 2011 IEEE International Conference on,pp.1945-7871

[8.]Yuedong Xu, Eitan Altman2, Rachid El-Azouzi, Salah Eddine Elayoubi3, Majed Haddad. QoE Analysis of Media Streaming in Wireless Data Networks [WWW]. Available at: http://www-sop.inria.fr/members/Eitan.Altman/PAPERS/qoe-ntkg-2012.pdf.

[9.]Takahiro Kawahara. Extended Discrete Autoregressive Model of Order One and Its Tractable Fitting Algorithm. Dissertation.Kyoto, 2007. Kyoto University. Publication - Kyoto University. 25p.

[10.]Alexis Michael Tourapis. H.264/14496-10 AVC Reference Software Manual [WWW]. Available at: http://iphome.hhi.de/suehring/tml/

[11.]Peak Signal-to-Noise Ratio as an Image Quality Metric. [WWW]. Available at : http://www.ni.com/white-paper/13306/en

[12.]Dmitri Moltchanov. Service quality in P2P streaming systems. Computer Science Review 5(Nov.2011)4,pp.319-340

[13.]Zhou Weihui. Analysis of Discrete-time Queue and its Application in Computer Network. Thesis. Guangzhou, 2004.Sun Yat-sen University.Publication - Sun Yat-sen University. 77p

[14.]FFmpeg. http://www.ffmpeg.org/about.html

[15.]Mean Opinion Score. http://en.wikipedia.org/wiki/Mean_opinion_score

[16.]AVS. http://www.avs.org.cn/

[17.] Video Codecs. http://baike.baidu.com/view/89060.; http://en.wikipedia.org/wiki/Video_codec

[18.] ITU-T : http://en.wikipedia.org/wiki/ITU-T

[19.]Video Encoding. http://baike.baidu.com/view/746807.htm?fromId=572258

[20.]Video Compression. http://en.wikipedia.org/wiki/Data_compression#Video

[21.]Drop Tail . http://www.cnblogs.com/cane004/archive/2009/11/30/DropTail.html

[22.]Random Early Detection. http://en.wikipedia.org/wiki/Random_early_detection

[23.]Walter Cerroni. Active Queue Management. [WWW]. Available at :http://www.sis.pitt.edu/~wcerroni/Lecture08.pdf

[24.] Set Top Box. http://www.which.co.uk/technology/tv-and-dvd/reviews/freeview-and-freesat-set-top-boxes/page/features-explained/#ixzz1rQIGU6hU

[25.] ITV. http://www.itv.com/tvguide/

[26.]Streaming Media. http://en.wikipedia.org/wiki/Streaming_media

[27.] Video Live Broadcasting. http://baike.baidu.com/view/1635669.htm

[28.] Video Streaming . http://www.um.u-tokyo.ac.jp/publish_db/2000dm2k/english/01/01-13.html

[29.] Streaming Technique . http://blog.csdn.net/freedom0203/article/details/2295969